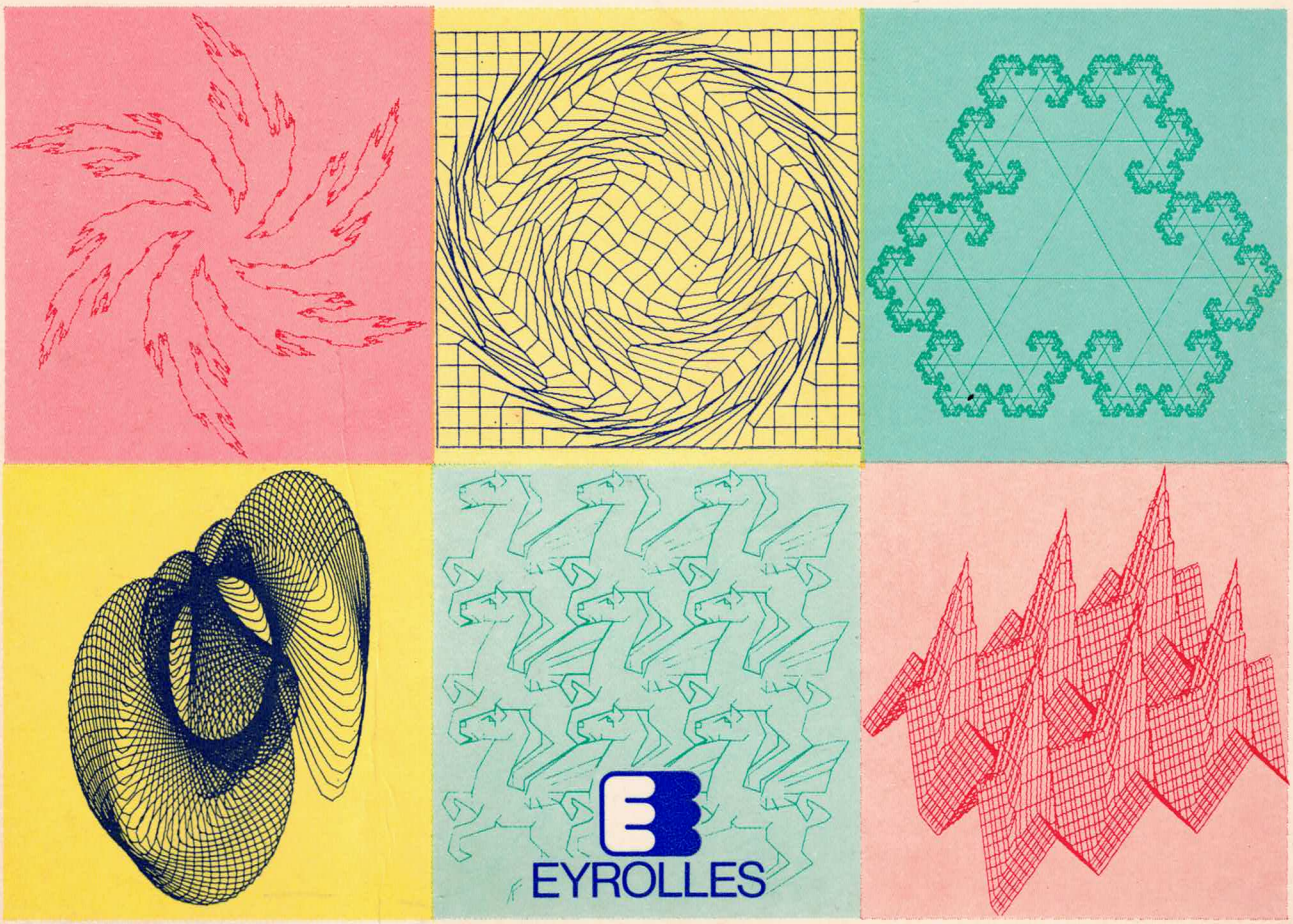
DESSINS GÉOMÉTRIQUES ET ARTISTIQUES AVEC VOTRE MICRO-ORDINATEUR

Jean-Paul DELAHAYE



EYROLLES

CHEZ LE MEME EDITEUR

COHEN	- *L'Apple animé 3D* - 184 p. ; 1984.
DELANNOY	- *Apprendre à programmer en Basic* - 272 p. ; 1984.
MONTEIL	- *Premiers pas en Logo* - 104 p. ; 1984, (coll. Micro-ordinateurs).
O'MALLEY	- *25 programmes graphiques en Basic Microsoft* - 192 p. ; 1985,
REVERCHON et DUCAMP	- *Mathématiques sur micro-ordinateur.* tome 1 : Analyse - 260 p. ; 1985.
VULDY	- *Graphisme 3D sur votre micro-ordinateur* - 128 p. ; 1985, (coll. Micro-ordinateurs).
WEIDENFELD, MATHIEU et PEROLAT	- *Logo* - 160 p. ; 1984.
WEIDENFELD, BRUILLARD et PEROLAT	- *Aller plus loin en Logo* - 136 p. ; 1985.

DESSINS GÉOMÉTRIQUES ET ARTISTIQUES AVEC VOTRE MICRO-ORDINATEUR

par

Jean-Paul DELAHAYE

EYROLLES

61, boulevard Saint-Germain — 75005 Paris
1985

Si vous désirez être tenu au courant de nos publications, il vous suffit d'adresser votre carte de visite au :

Service « Presse », Éditions EYROLLES
61, Boulevard Saint-Germain,
75240 PARIS CEDEX 05

en précisant les domaines qui vous intéressent. Vous recevrez régulièrement un avis de parution des nouveautés en vente chez votre libraire habituel.

« La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40) ».

« Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal ».

© Éditions EYROLLES, 1985

Introduction

Réussir à faire de beaux dessins avec son ordinateur, ce n'est pas difficile. C'est ce que ce livre se propose de montrer.

Il y a deux façons de l'utiliser.

La première consiste à recopier les programmes correspondant aux dessins qui vous plaisent, ce qui vous permettra de les obtenir sur votre écran ou de les faire exécuter par votre traceur.

La seconde consiste à comprendre les programmes, ce qui est rendu aisé par les explications les accompagnant, et ce qui vous permettra de réaliser (en plus des dessins donnés en exemple) des dessins que vous inventerez vous-même. En effet, tous les programmes ont été conçus pour qu'on puisse obtenir des variantes très différentes à l'aide de petites modifications. Avec un peu de patience et d'imagination vous saurez comment modifier les programmes donnés et ainsi réaliser vos propres créations géométriques et artistiques.

Les connaissances requises en BASIC et en mathématiques sont nulles si vous optez pour la première méthode d'utilisation du livre. Elles ne sont pas bien élevées si vous choisissez la seconde (des rappels élémentaires, mais suffisants, sont donnés avec les explications des programmes).

Nous avons essayé de concevoir ce livre de telle façon que tout le monde puisse trouver à se satisfaire : il y a des programmes très simples et d'autres un peu plus élaborés. Les programmes ne sont jamais longs car nous sommes partis de l'idée que tous devaient pouvoir être saisis en une demi-heure au plus. Les idées de base sont rarement très compliquées bien que les résultats paraissent parfois d'une complexité inouïe.

Les algorithmes présentés le sont toujours le plus simplement possible, de plus ils sont rédigés pour que 8 K octets de mémoire vive soient suffisants. Cela a parfois de légers inconvénients : temps d'exécution longs, difficultés de cadrage. Les perfectionnements possibles des programmes (qui, bien sûr, les compliqueront) sont d'excellentes occasions d'exercices de programmation.

Le but fondamental de ce livre reste malgré tout de vous amener à exprimer votre créativité dans la réalisation de dessins géométriques et artistiques personnels.

Table des matières

INTRODUCTION VII

0. Comment utiliser les programmes 1

1. Polygones réguliers, étoiles, etc. 7
Le programme POLYGONES RÉGULIERS 8
Le programme ÉTOILES RÉGULIÈRES 12
Le programme COMPOSITION 1 15
Le programme COMPOSITION 2 22
Le programme JOLIGONES 28

2. Dessins à partir de données 35
Le programme CHEVAL 37
Les programmes LION, OISEAUX-POISSONS, SMURF 49

3. Dragons de papiers pliés 57
Le programme DRAGONS 59

4. Etoiles fractales 71
Le programme ÉTOILES FRACTALES 72

5. Courbes 81
Le programme COURBES ORBITALES 81
Le programme COURBES TOURNANTES 91
Le programme COURBES SPIRALES 99

6. Dessins linéaires 103
Le programme BIPARTI COMPLET 104
Le programme LINÉAIRES MODULO 109
Le programme LINÉAIRES BATONS 115

7. Fractales simples 121
Le programme FRACTALES SIMPLES 121
Le programme FRACTALES SIMPLES ARRONDIES 139
Le programme FRACTALES SIMPLES DÉFORMÉES 151

8. Quadrillages élastiques 163
Le programme QUADRILLAGES ÉLASTIQUES 163

9. Dessiner des surfaces 173
Le programme SURFACES 173

10. La troisième dimension 195
Le programme D3 DATA 197
Le programme D3 CUBE 203
Le programme D3 STRUCTURES 215

Annexe. — Exemples d'adaptations de programmes 237

Références bibliographiques 243

Note :
Les programmes des chapitres 1, 5, 6 et 8 sont les plus faciles à utiliser et à maîtriser.

0 Comment utiliser les programmes

I. — Utilisation possible des programmes sur tout micro-ordinateur

Tous les programmes de dessins présentés ici ont été rédigés dans un langage BASIC standard (MICROSOFT). Les principes utilisés pour les écrire sont très simples et donc, moyennant de légères adaptations indiquées dans ce chapitre (voir § 4) :

Tous les programmes peuvent être utilisés sur tous les micro-ordinateurs ayant un minimum de possibilités graphiques, soit pour obtenir le dessin sur l'écran, soit pour commander un traceur connecté au micro-ordinateur.

Les listings donnés correspondent aux programmes pour le CANON X 07 connecté à son traceur X 710. Pour ceux qui possèdent un tel système (valant moins de 5 000 F) aucune adaptation n'est nécessaire, il suffit de recopier les programmes et d'appuyer sur RUN !

II. — Début des programmes

Les principes utilisés pour la rédaction des programmes sont les suivants :

- à la ligne 10 est indiqué un nom de programme (le signe ' est équivalent à REM)
- à la ligne 50 sont placés :
  - les ordres d'initialisation du traceur (déplacement du papier, fixation de l'origine)
  - la variable NP. Pour nous NP = 480 car le traceur, que nous utilisons pour la réalisation des dessins, comporte une largeur de 480 pas élémentaires. *Pour adapter les programmes à d'autres systèmes, vous devez fixer NP = nombre de pas élémentaires de votre traceur ou NP = (le plus petit des deux nombres donnant la résolution de votre écran) — 1.*

    Exemple :

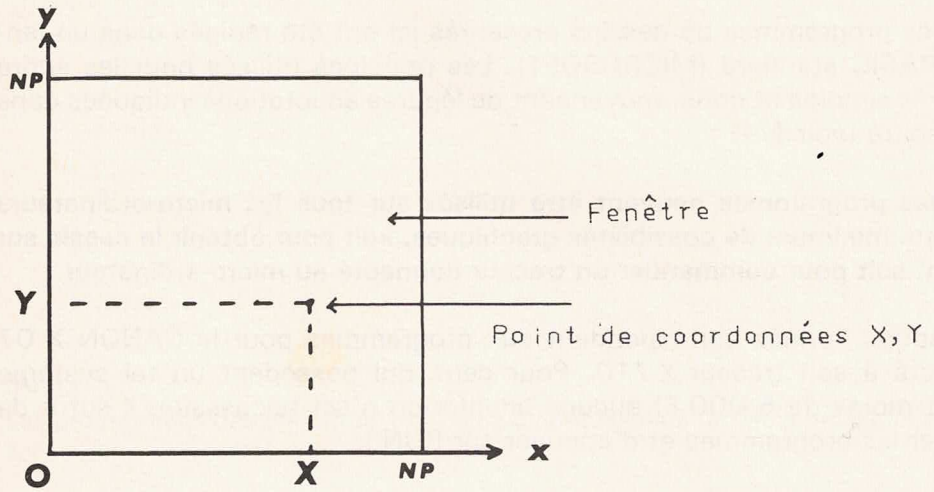
    NP = 255 pour un ordinateur DAI qui dispose d'une résolution de 336 x 256 ; NP = 175 pour le Sinclair ZX Spectrum qui dispose d'une résolution de 256 x 176.
  - le calcul de PI = 3,14159 ... = 4 * ATN (1).

    (sur certains ordinateurs, ce calcul est inutile).

III. — Fenêtres utilisées

La plupart des programmes réalisent le dessin dans un carré (appelé fenêtre) dont chaque point est repéré par 2 entiers (appelés coordonnées)

X compris entre O et NP,
Y compris entre O et NP.



Parfois une fenêtre plus grande est utilisée :

X compris entre 0 et NP,
Y compris entre 0 et 2 * NP,

ou

X compris entre 0 et NP,
Y compris entre — NP et + NP.

Si vous rencontrez des problèmes de fenêtres ou d'images déformées, voici comment faire.

Dans les ordres graphiques, vous remplacez X, Y par :

X, Y + DE : pour faire déplacer le dessin de DE pas élémentaires vers le haut (en prenant DE < 0 le dessin se déplacera vers le bas).

X, CE * Y : pour aplatir le dessin en largeur, avec CE < 1 (par exemple CE = .8, CE = .9).

CE * X, Y : pour aplatir le dessin en hauteur avec CE > 1.

CE * X, CE * Y : pour obtenir un dessin plus petit avec CE < 1.

CE * X, CE * Y : pour obtenir un dessin plus grand avec CE > 1.

X, NP — Y : pour retourner le dessin. Beaucoup d'ordinateurs fixent l'origine de l'écran en haut à gauche (alors que nous utilisons toujours une origine en bas à gauche de la feuille), si c'est le cas de votre ordinateur vous avez besoin de retourner le dessin (quoique pour certaines figures géométriques, cela soit sans importance).

Les paramètres DE, CE que nous vous suggérons d'utiliser doivent être fixés en début de programme, par exemple à la ligne 50.

De manière à alléger les programmes qui parfois tournent lentement (à cause de l'utilisation des fonctions trigonométriques et de grandes quantités de calculs), nous avons délibérément omis toute procédure de contrôle (vérification par exemple que le segment que l'on trace ne sort pas de la fenêtre choisie). De même aucune technique de centrage automatique n'a été utilisée. Celui qui veut se servir des programmes sans chercher à les comprendre ne sera pas gêné car les programmes proposés réalisent des dessins bien positionnés. Celui qui, au contraire, veut approfondir les programmes et créer avec eux ses propres images, grâce aux nombreux exemples, pourra facilement apprendre à choisir les paramètres pour obtenir des dessins bien cadrés et bien centrés.

IV. — Traduction des ordres graphiques utilisés

De façon à faciliter l'adaptation à tout système de ces programmes, *deux ordres graphiques seulement* sont utilisés dans ce livre :

ordre 1 : se déplacer au point de coordonnées X, Y

ordre 2 : tracer un segment de droite entre le point où on est placé et le point de coordonnées X, Y

Pour le CANON X 07 connecté à son traceur X 710 :

ordre 1 : LPRINT ''M'' ; X ; '','' ; Y

ordre 2 : LPRINT ''D'' ; X ; '','' ; Y

Pour les ordinateurs suivants : BBC Micro computer ; Acorn Atom ; Grundy New Brain ;

la traduction est la suivante :

ordre 1 : MOVE X, Y

ordre 2 : DRAW X, Y
(ou DRAW — [X, Y])

Certains ordinateurs ne possèdent pas d'ordre équivalent à l'ordre 1 mais par contre, disposent d'un autre ordre :

ordre 3 : tracer un segment de droite entre les points de coordonnées (X1, Y1) et (X2, Y2).

Avec ce seul ordre, on peut remplacer l'ordre 1 et l'ordre 2 de la manière suivante :

ordre 1 : XW = X : YW = Y

ordre 2 : ordre 3 (XW, YW) — (X, Y) : XW = X : YW = Y

(un exemple de traduction de programme utilisant cette méthode est donné en annexe : PROGRAMME ÉTOILES FRACTALES pour Apple II - moniteur).

Cet ordre 3 s'écrit

(a) ordre 3 : LINE (X1, Y1) — (X2, Y2)

sur les ordinateurs EPSON HX 20 ; Sharp PC 1500 ; T 07 ; M 05 ; Sharp MZ 80-B ; NEC PC 8 001 ; DRAGON G4 ; Standard MSX ;

(b) ordre 3 : DRAW (X1, Y1) — (X2, Y2)

sur les ordinateurs DAI ; Sinclair ZX Spectrum ; Tandy color computer ;

(c) ordre 3 : PLOT (X1, Y1) — (X2, Y2)

sur les ordinateurs Wang ;

(d) ordre 3 : HPLOT X1, Y1 TO X2, Y2
sur Apple II (dessin en haute résolution sur le moniteur).

Les utilisateurs de l'Oric et de l'Atmos doivent faire attention car l'ordre DRAW dont ils disposent travaille en coordonnées relatives (au lieu des coordonnées absolues). Ils doivent utiliser la traduction suivante :

ordre 1 : XW = X, YW = Y

ordre 2 : DRAW X − XW, Y − YW : XW = X : YW = Y

Des exemples de traduction sont donnés dans l'annexe.

Dans nos programmes nous n'utilisons pas de couleurs : il ne tient qu'à vous d'introduire (si vous en disposez) des instructions de coloration. De même, certains programmes peuvent être simplifiés si vous disposez d'instructions graphiques plus élaborées comme CIRCLE.

V. – Le IF... THEN... ELSE

Si votre ordinateur n'accepte pas l'ordre :

(n° ligne) *IF* condition *THEN* instruction 1 *ELSE* instruction 2

vous pouvez dans les programmes qui contiennent une telle ligne écrire :

(n° ligne)	*IF* condition *THEN* instruction 1 : *GOTO* (n° ligne + 2)
(n° ligne + 1)	instruction 2
(n° ligne + 2)	*REM*

Exemple :

```
200   IF A = 0 THEN A = B ELSE A = A * C
```

devient :

```
200   IF A = 0 THEN A = B : GOTO 202
201   A = A * C
202   REM
```

VI. – Programmes de base et modifications

De façon à éviter de recopier plusieurs fois des programmes peu différents, nous avons rédigé un nombre limité *de programmes de base,* dont quelques

lignes seulement sont ensuite changées ou ajoutées pour obtenir des dessins différents.

Notre présentation est toujours la même :

Programme de base

```
IO ' TITRE
. . .
. . .                  } corps du programme
. . .
. . .
```

Modifications pour le dessin X

```
IO ' TITRE
. . .
. . .                  } lignes changées ou ajoutées
. . .                    par rapport au programme de base
. . .
```

Tous les programmes comportent des paramètres. Le plus souvent en modifiant un ou plusieurs d'entre eux, on obtient des dessins très différents les uns des autres.

Pour tirer un profit maximum de ce livre, vous devez essayer de comprendre comment fonctionnent les programmes et les variantes proposées (les explications données avec chaque programme vous y aideront).

Vous pourrez alors créer des dessins totalement nouveaux et inattendus, vous aurez franchi la frontière de l'informatique et vous vous trouverez dans le domaine de l'art graphique.

1 Polygones réguliers, étoiles etc.

Dans ce premier chapitre, nous commençons par deux programmes très élémentaires (« POLYGONES RÉGULIERS » et « ÉTOILES RÉGULIÈRES ») qui permettent de tracer quelques figures géométriques de base. Ces programmes ont pour but de vous rappeler (ou de vous faire connaître) quelques formules simples et de vous montrer comment les utiliser. Les paramètres introduits ne sont pas très nombreux, entraînez-vous à les maîtriser en faisant divers essais.

Suivent alors deux programmes (« COMPOSITION 1 » et « COMPOSITION 2 ») qui utilisent le principe des polygones et des étoiles. Là encore, tout reste assez simple et on arrive facilement à contrôler les divers paramètres. Les multiples variantes que l'on peut réaliser sont déjà assez intéressantes.

Le dernier programme du chapitre (« JOLIGONES ») est très classique et, lui aussi, offre la possibilité d'une infinité de variations surprenantes et fascinantes.

Le programme POLYGONES RÉGULIERS

En plaçant K points également espacés sur un cercle on obtient ce qu'on appelle un polygone régulier. Le programme proposé ici permet de tracer n'importe quel polygone régulier.

Ligne 50 :
Initialisation. Voir le chapitre 0.

Ligne 100

On fixe les différents paramètres qui sont :

K : nombre de sommets du polygone régulier.
Ici K = 4 ; on doit donc obtenir un carré.

CX, CY : coordonnées du centre C du cercle sur lequel sont disposés les sommets. Ici CX = NP/2, CY = NP/2 ce qui correspond exactement au centre de la fenêtre (voir chapitre 0).

R : rayon du cercle.

AD : angle mesuré en radians(*) du vecteur CS avec l'horizontale, S étant le sommet où commence le dessin.

Ligne 200

Début de la boucle principale.

Lignes 210, 220

On calcule les coordonnées X et Y du I-ième sommet du polygone, puis on prend les parties entières. Il n'est pas nécessaire de savoir démontrer les formules donnant X et Y mais regardez-les bien, nous les réutiliserons.

Lignes 250, 260

Lorsque I = 0, on se déplace sans rien tracer jusqu'au sommet numéro 0.

Lorsque I>0, on se déplace en traçant un segment jusqu'au sommet numéro I. Comme la pointe était placée sur le sommet numéro I-1, on obtient un côté du polygone.

Lignes 300, 500

Fin de la boucle principale. Fin du programme.

(*) 2 PI radians = 360 °, donc PI radians = 180 °
PI/2 radians = 90 °, PI/3 radians = 60 °, PI/4 radians = 45 °, PI/6 radians = 30 °.

Programme de base « POLYGONES RÉGULIERS »

```
10 'POLYGONES REGULIERS
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 K=4:CX=NP/2:CY=NP/2:R=NP*.45:AD=PI/4
200 FOR I=0 TO K
210 X=CX+R*COS(2*I*PI/K+AD):X%=INT(X)
220 Y=CY+R*SIN(2*I*PI/K+AD):Y%=INT(Y)
250 IF I=0 THEN LPRINT"M";X%;",";Y%
260 IF I>0 THEN LPRINT"D";X%;",";Y%
300 NEXT I
500 END
```

Le dessin 1 donne bien le carré attendu. Que se passerait-il si on prenait AD = 0 à la place de AD = PI/4 ? Les dessins 2, 3, 4, 5, 6 montrent divers essais ; faites-en d'autres...

Modifications pour le dessin 1

```
10 'POLYGONES REGULIERS
```

Modifications pour le dessin 2

```
10 'POLYGONES REGULIERS
100 K=3:CX=NP/2:CY=NP/2:R=NP*.45:AD=0
```

Modifications pour le dessin 3

```
10 'POLYGONES REGULIERS
100 K=3:CX=NP/2:CY=NP/2:R=NP*.45:AD=PI/2
```

Modifications pour le dessin 4

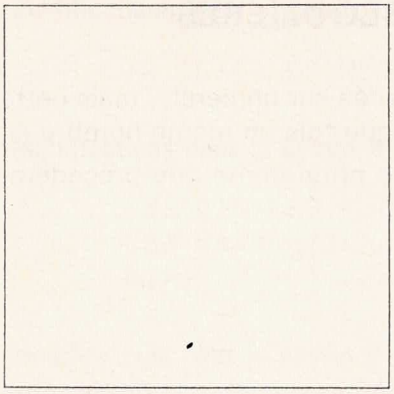
```
10 'POLYGONES REGULIERS
100 K=5:CX=NP/2:CY=NP/2:R=NP*.45:AD=PI/2
```

Modifications pour le dessin 5

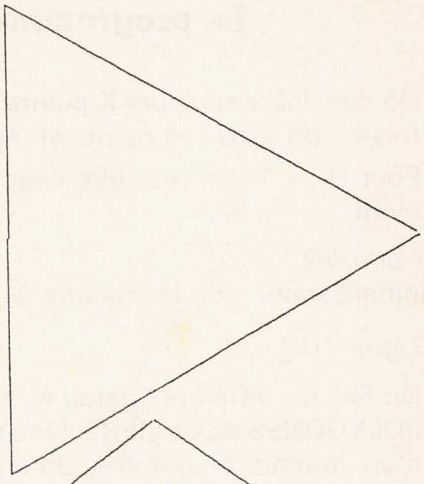
```
10 'POLYGONES REGULIERS
100 K=8:CX=NP/2:CY=NP/2:R=NP*.5:AD=PI/8
```

Modifications pour le dessin 6

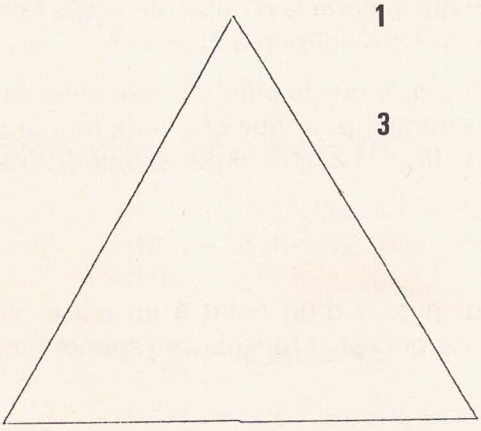
```
10 'POLYGONES REGULIERS
100 K=20:CX=NP/2:CY=NP/2:R=NP*.4:AD=0
```



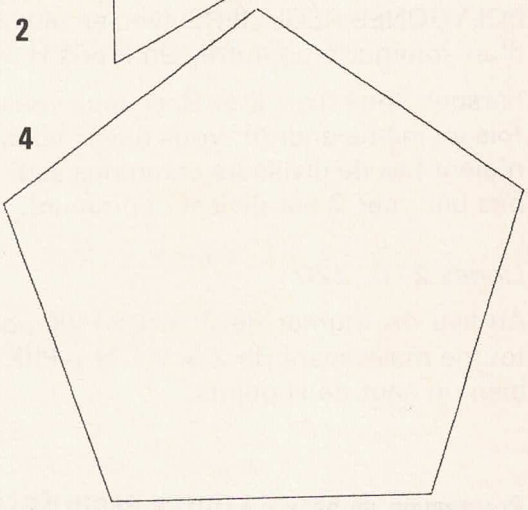
1



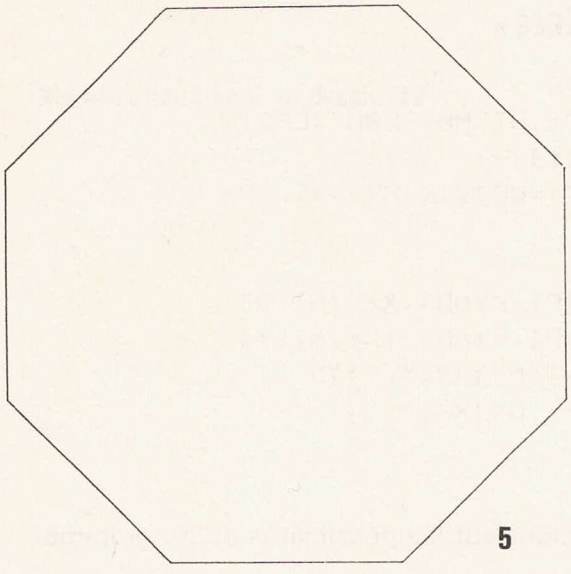
2



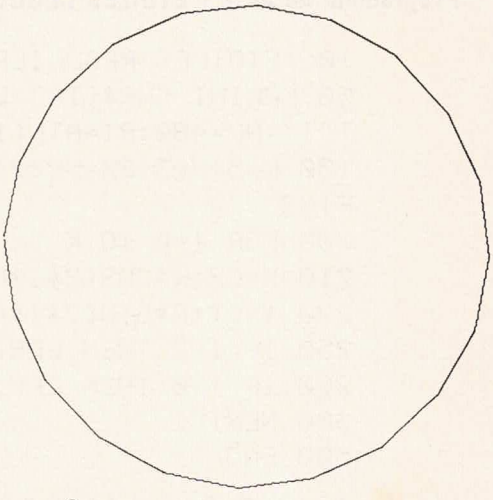
3



4



5



6

Le programme ÉTOILES RÉGULIÈRES

On considère toujours K points également espacés sur un cercle, mais cette fois-ci, on joint ces points en en sautant à chaque fois un même nombre H.

Pour H = 1, on retrouve exactement le même programme que précédemment.

Ligne 50

Initialisation. Voir le chapitre 0.

Ligne 100

On fixe les différents paramètres. Ce sont les mêmes que pour le programme POLYGONES RÉGULIERS avec en plus H qui indique le nombre de sauts faits d'un sommet à un autre. On a pris H = 3. Que donnerait H = 2 ?

Lorsque vous fixez K et H, si vous voulez éviter que la pointe passe plusieurs fois au même endroit, vous devez vous arranger pour que ces deux nombres n'aient pas de diviseurs communs sauf 1. (K = 12, H = 9 par exemple n'est pas bon, car 3 est diviseur commun).

Lignes 210, 220

Au lieu de tourner de 2 * I *PI/K pour passer d'un point à un autre, on tourne maintenant de 2 * I * H * PI/K, ce qui est H fois plus et donne donc bien un saut de H points.

Programme de base « ÉTOILES RÉGULIÈRES »

```
10 'ETOILES REGULIERES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 K=5:H=3:CX=NP/2:CY=NP/2:R=NP*.45:AD=
PI/2
200 FOR I=0 TO K
210 X=CX+R*COS(2*I*H*PI/K+AD):X%=INT(X)
220 Y=CY+R*SIN(2*I*H*PI/K+AD):Y%=INT(Y)
250 IF I=0 THEN LPRINT"M";X%;",";Y%
260 IF I>0 THEN LPRINT"D";X%;",";Y%
300 NEXT I
500 END
```

Les dessins 7, 8, 9, 10, 11 et 12 montrent les possibilités du programme.

Modifications pour le dessin 7

```
10 'ETOILES REGULIERES
```

Modifications pour le dessin 8

```
10 'ETOILES REGULIERES
100 K=7:H=3:CX=NP/2:CY=NP/2:R=NP*.45:AD=
PI/2
```

Modifications pour le dessin 9

```
10 'ETOILES REGULIERES
100 K=20:H=9:CX=NP/2:CY=NP/2:R=NP*.45:AD
=PI/2
```

Modifications pour le dessin 10

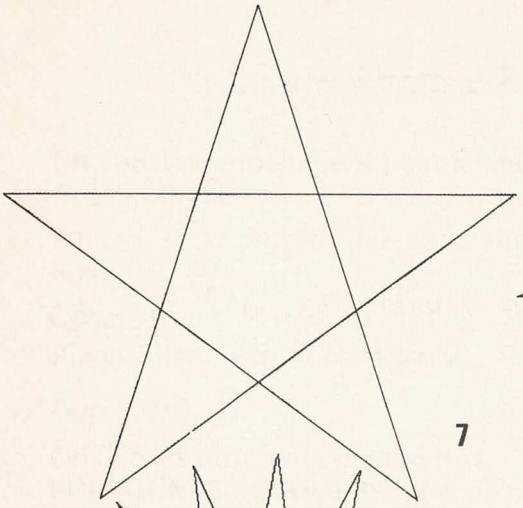
```
10 'ETOILES REGULIERES
100 K=20:H=7:CX=NP/2:CY=NP/2:R=NP*.45:AD
=PI/2
```

Modifications pour le dessin 11

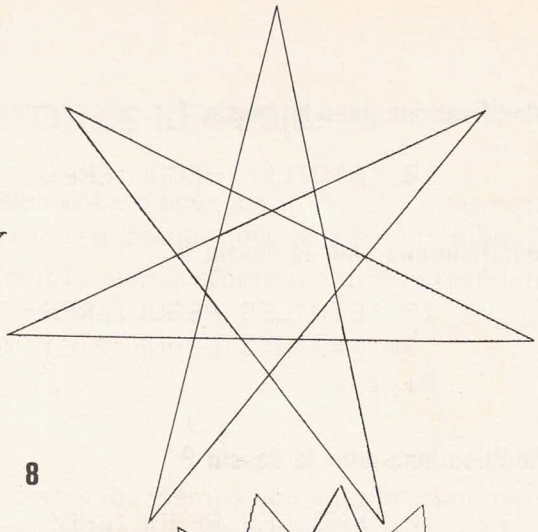
```
10 'ETOILES REGULIERES
100 K=51:H=20:CX=NP/2:CY=NP/2:R=NP*.45:A
D=PI/2
```

Modifications pour le dessin 12

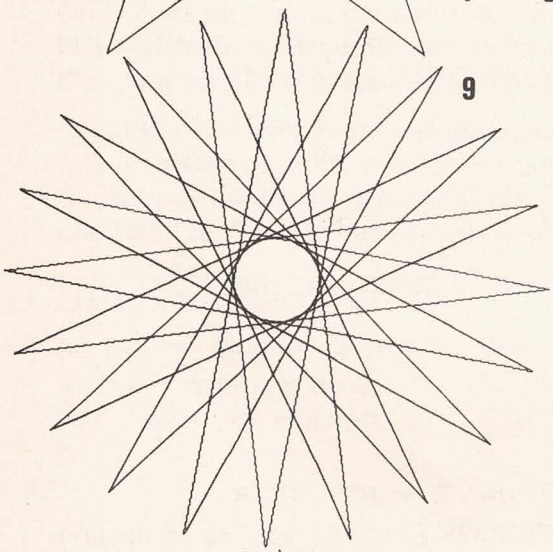
```
10 'ETOILES REGULIERES
100 K=51:H=25:CX=NP/2:CY=NP/2:R=NP*.45:A
D=PI/2
```



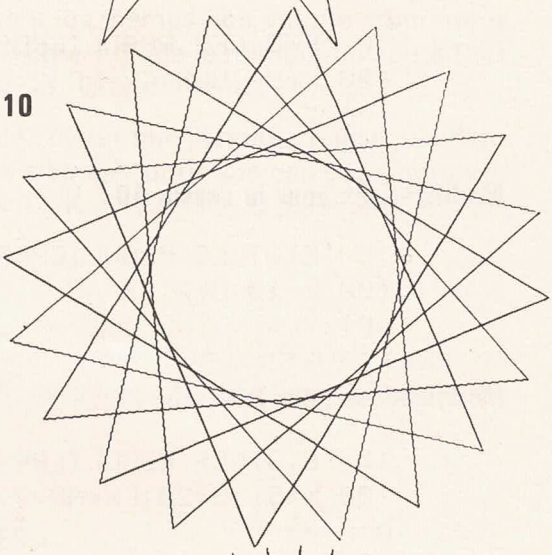
7



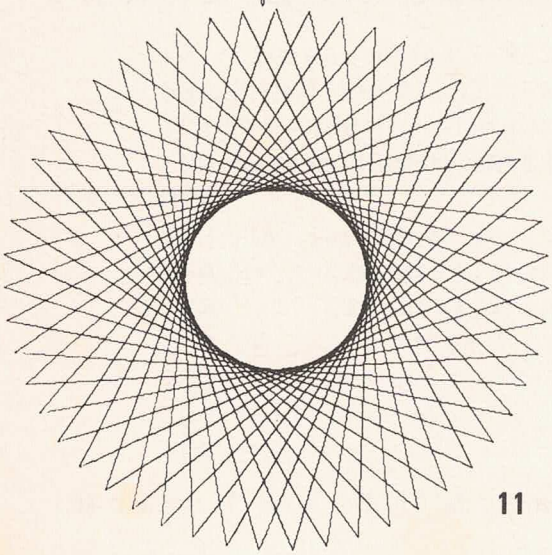
8



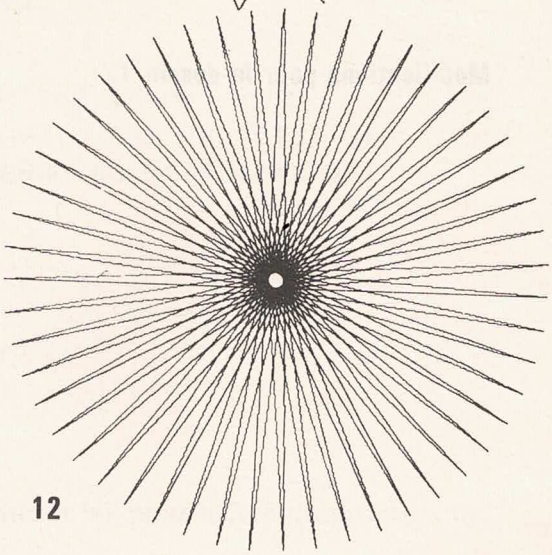
9



10



11



12

Le programme COMPOSITION 1

L'idée de ce programme est assez simple : on va dessiner K1 étoiles régulières dont les centres seront également espacés sur un cercle (autrement dit dont les centres seront les sommets d'un polygone régulier).

Ligne 50

Initialisation. Voir le chapitre 0.

Ligne 100

On fixe les différents paramètres qui sont :

K1 : nombre d'étoiles,

DX, DY : coordonnées du centre C du cercle sur lequel sont les centres des étoiles,

R1 : rayon de ce cercle,

A1 : angle mesuré en radians du vecteur CS avec l'horizontale, S étant le centre de la première étoile dessinée.

Ligne 110

On fixe les paramètres qui déterminent la forme des étoiles :

K : nombre de sommets des étoiles,

H : nombre de sauts dans le dessin d'une étoile (voir programme ÉTOILES RÉGULIÈRES),

R : rayon des cercles qui portent les sommets des étoiles,

AD : angle (mesuré en radians) du vecteur correspondant au premier point de chaque étoile (voir le programme POLYGONES RÉGULIERS).

Ligne 200

Début de la boucle principale ; à chaque parcours de la boucle, une étoile est dessinée.

Lignes 210 - 220

Calcul des coordonnées du centre de l'étoile numéro I 1. On remarquera que les formules utilisées sont celles qui dans le programme POLYGONES RÉGULIERS, déterminent les coordonnées des sommets.

Ligne 250

Appel du sous-programme.

Lignes 300 - 500

Fin de la boucle principale ; fin du programme.

Lignes 5000 à 5500

Sous-programme construit sur le modèle du programme ÉTOILES RÉGULIÈRES et qui dessine une étoile dont les paramètres sont K, H, CX, CY, R, AD.

Programme de base « COMPOSITION 1 »

```
10 'COMPOSITION 1
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 K1=5:DX=NP/2:DY=NP/2:R1=NP*.27:A1=PI
/2
110 K=25:H=12:R=NP*.22:AD=PI/2
200 FOR I1=0 TO K1-1
210 CX=DX+R1*COS(2*PI*I1/K1+A1)
220 CY=DY+R1*SIN(2*PI*I1/K1+A1)
250 GOSUB 5000
300 NEXT I1
500 END
5000 'SOUS-PROGRAMME ETOILES REGULIERES
5200 FOR I=0 TO K
5210 X=CX+R*COS(2*I*H*PI/K+AD):X%=INT(X)
5220 Y=CY+R*SIN(2*I*H*PI/K+AD):Y%=INT(Y)
5250 IF I=0 THEN LPRINT"M";X%;",";Y%
5260 IF I>0 THEN LPRINT"D";X%;",";Y%
5300 NEXT I
5500 RETURN
```

Le dessin 13 correspond aux données du programme de base : K1 = 5 étoiles régulières, ayant chacune K = 25 branches.

Quand les différentes étoiles se rejoignent, on obtient une sorte de broderie. Cela se produit quand R est plus grand que R1. Le dessin 14 fournit un exemple.

Si on prend R1 = R2 et K assez grand avec H = 1, on obtient des cercles passant par le centre du dessin. C'est ce qu'on a fait pour le dessin 15.

Les dessins 16, 17, 18 et 19 montrent encore d'autres possibilités. Essayez de comprendre pourquoi on obtient de telles figures ; recherchez d'autres jeux de paramètres intéressants.

Modifications pour le dessin 13

```
10 'COMPOSITION 1
```

Modifications pour le dessin 14

```
10 'COMPOSITION 1
100 K1=6:DX=NP/2:DY=NP/2:R1=NP*.2:A1=0
110 K=24:H=11:R=NP*.3:AD=0
```

Modifications pour le dessin 15

```
10 ' COMPOSITION 1
100 K1=40:DX=NP/2:DY=NP/2:R1=NP*.25:A1=P
I/2
110 K=80:H=1:R=NP*.25:AD=PI/2
```

Modifications pour le dessin 16

```
10 ' COMPOSITION 1
100 K1=10:DX=NP/2:DY=NP/2:R1=NP*.35

110 K=10:H=3:R=NP*.15:AD=0
```

Modifications pour le dessin 17

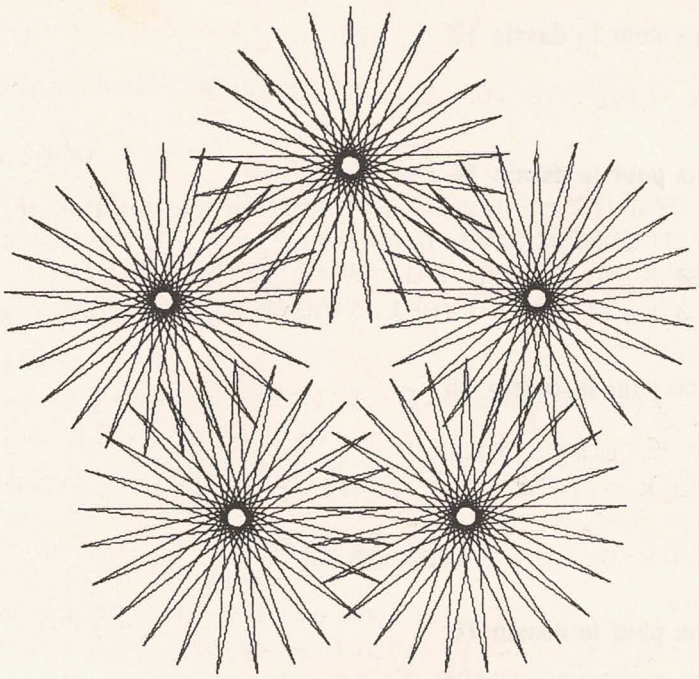
```
10 'COMPOSITION 1
100 K1=63:DX=NP/2:DY=NP/2:R1=NP*.15:A1=0
110 K=4:H=1:R=NP*.35:AD=0
```

Modifications pour le dessin 18

```
10 ' COMPOSITION 1
100 K1=25:DX=NP/2:DY=NP/2:R1=NP*.1:A1=PI
/2
110 K=5:H=2:R=NP*.4:AD=PI/2
```

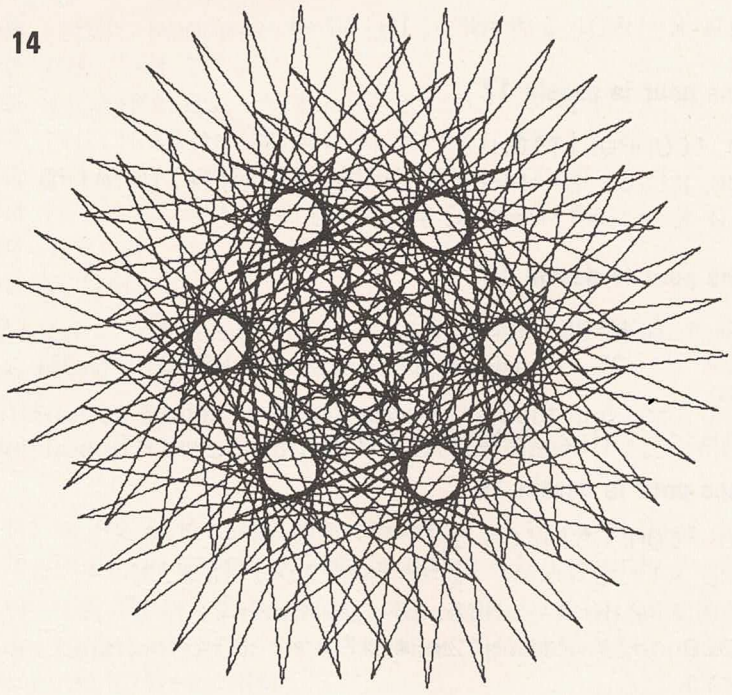
Modifications pour le dessin 19

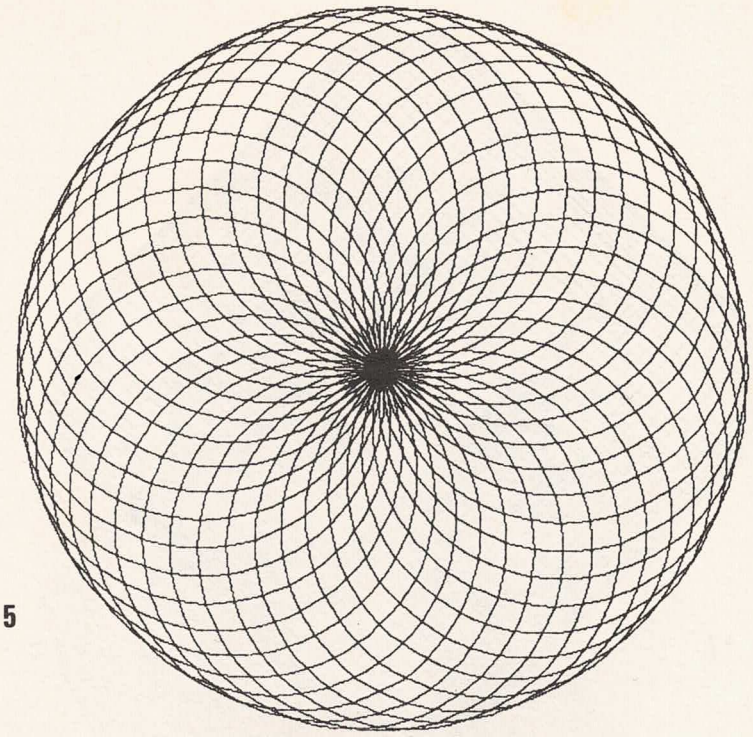
```
10 'COMPOSITION 1
100 K1=99:DX=NP/2:DY=NP/2:R1=NP*.25:A1=0
110 K=7:H=3:R=NP*.25:AD=PI/2
5220 Y=CY+R*SIN(2*I*H*PI/K+AD):Y%=INT(1.
5*Y)
```



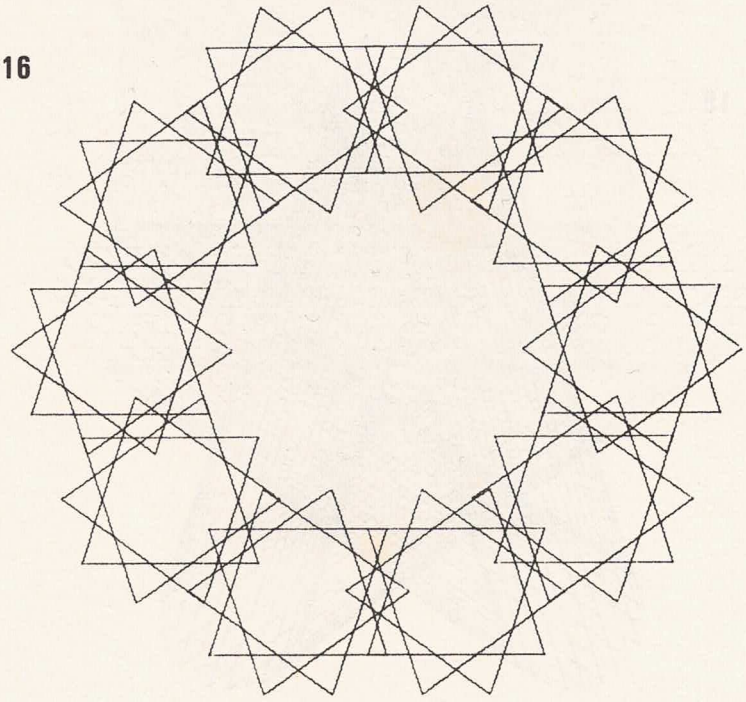
13

14

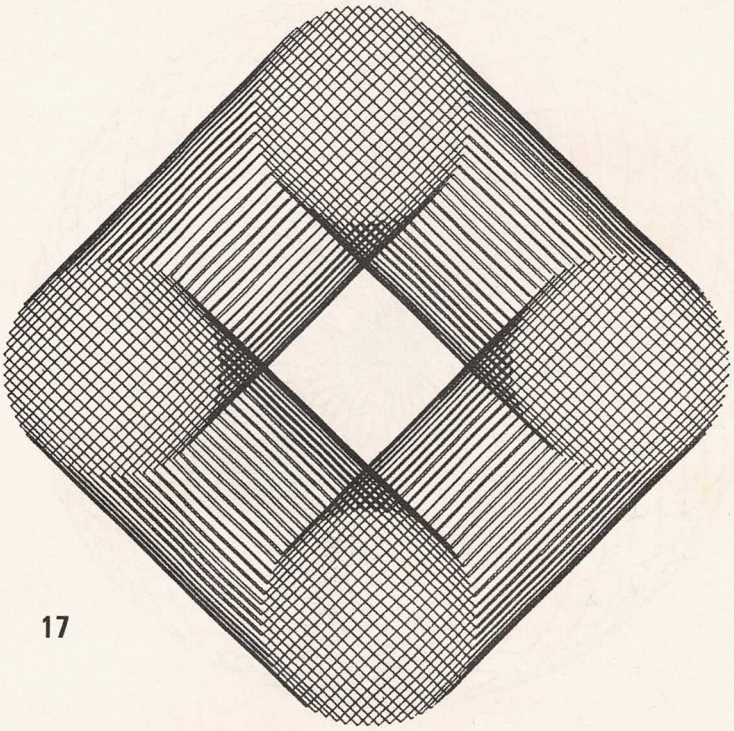




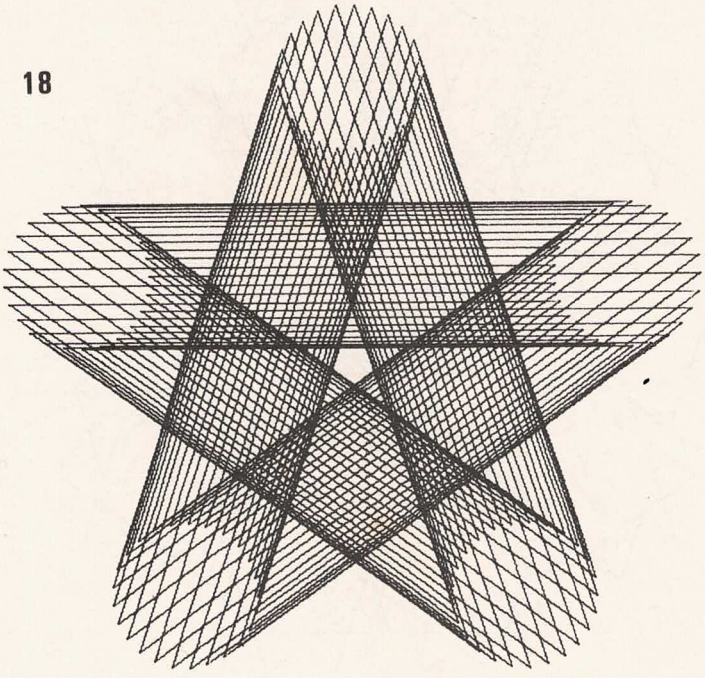
15



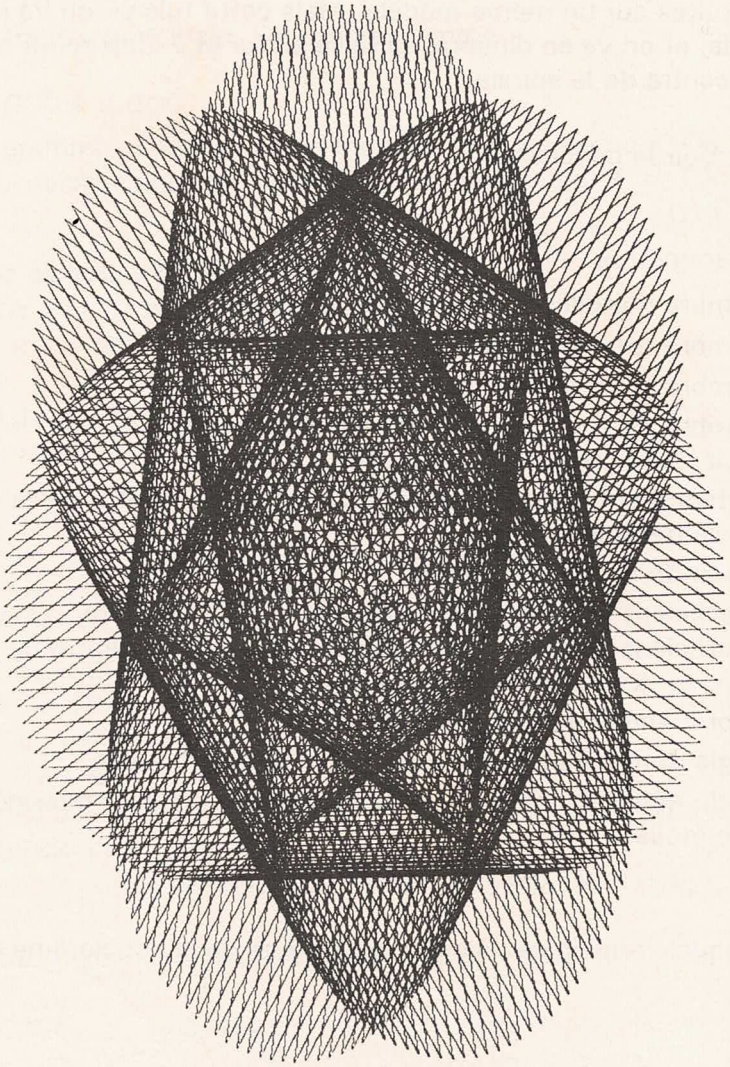
16



17



18



19

Le programme COMPOSITION 2

Comme dans le programme précédent, on va dessiner des étoiles régulières toutes construites sur un même modèle. Mais cette fois-ci, on va les placer sur une spirale, et on va en diminuer la taille au fur et à mesure qu'on se rapprochera du centre de la spirale.

Ligne 50

Initialisation. Voir le chapitre 0.

Lignes 100, 110

On fixe les paramètres.

K1 : nombre d'étoiles à chaque tour de la spirale,

N : nombre total d'étoiles,

K : nombre de sommets de chaque étoile,

H : nombre de sauts dans le dessin d'une étoile (voir le programme ÉTOILES RÉGULIÈRES),

R1 : distance entre le centre de la spirale et le centre de la première étoile dessinée,

R : rayon de la première étoile,

RR : coefficient de réduction d'une étoile à la suivante, c'est aussi le coefficient de réduction de la distance entre le centre de la spirale et le centre des étoiles successives,

DX, DY : coordonnées du centre de la spirale,

A1 : angle déterminant la position de la première étoile,

AD : angle déterminant la position du premier sommet dessiné de chaque étoile.

Ligne 200

Début de la boucle principale ; à chaque parcours de la boucle, une étoile est dessinée.

Ligne 210

Calcul de R2, la distance entre le centre de la spirale et le centre de l'étoile numéro I 1.

Calcul de R3, le rayon du cercle de l'étoile numéro I 1.

Lignes 220, 230

Calcul des coordonnées du centre de l'étoile numéro I 1.

Ligne 250

Appel du sous-programme.

Lignes 300, 500

Fin de la boucle principale. Fin du programme.

Lignes 5 000 à 5 500

Sous-programme construit sur le modèle du programme ÉTOILES RÉGULIÈRES et qui dessine une étoile dont les paramètres sont K, H, CX, CY, R3, AD.

Programme de base « COMPOSITION 2 »

```
10 'COMPOSITION 2
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 K1=8:N=32:K=16:H=5:R1=NP*.36:R=NP*.1
4:RR=.9
110 DX=NP/2:DY=NP/2:A1=0:AD=0
200 FOR I1=0 TO N
210 R2=R1*RR^I1:R3=R*RR^I1
220 CX=DX+R2*COS(2*PI*I1/K1+A1)
230 CY=DY+R2*SIN(2*PI*I1/K1+A1)
250 GOSUB 5000
260 IF I>0 THEN LPRINT"D";X%;",";Y%
300 NEXT I1
500 END
5000 'SOUS-PROGRAMME ETOILES REGULIERES
5200 FOR I=0 TO K
5210 X=CX+R3*COS(2*I*H*PI/K+AD):X%=INT(X
)
5220 Y=CY+R3*SIN(2*I*H*PI/K+AD):Y%=INT(Y
)
5250 IF I=0 THEN LPRINT"M";X%;",";Y%
5260 IF I>0 THEN LPRINT"D";X%;",";Y%
5300 NEXT I
5500 RETURN
```

Les dessins 20 et 21 sont des utilisations immédiates du programme. Pour les dessins 22 et 23 on a pris R1 = 0 ce qui donne des étoiles de même centre, de tailles décroissantes. Pour les dessins 24 et 25, on a pris R plus grand que R1.

Modifications pour le dessin 20

```
10 'COMPOSITION 2
```

Modifications pour le dessin 21

```
10 ' COMPOSITION 2
100 K1=10:N=30:K=8:H=3:R1=NP*.35:R=NP*.1
5:RR=.85
```

Modifications pour le dessin 22

```
10 ' COMPOSITION 2
100 K1=10:N=10:K=18:H=7:R1=NP*.0:R=NP*.5
:RR=.80
```

Modifications pour le dessin 23

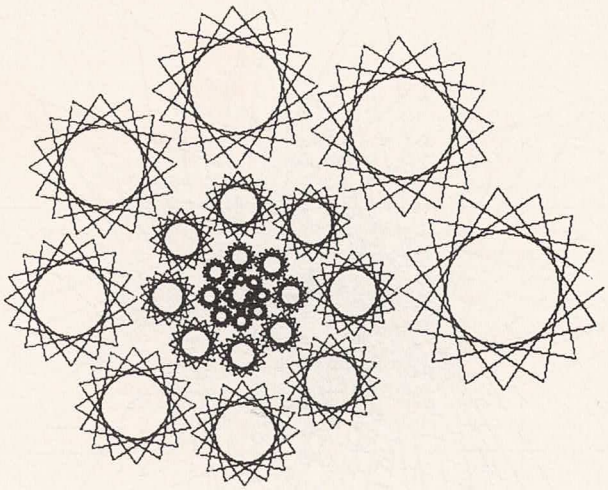
```
10 'COMPOSITION 2
100 K1=10:N=10:K=21:H=10:R1=NP*.0:R=NP*.
5:RR=.75
```

Modifications pour le dessin 24

```
10 'COMPOSITION 2
100 K1=28:N=56:K=7:H=3:R1=NP*.15:R=NP*.3
5:RR=.95
```

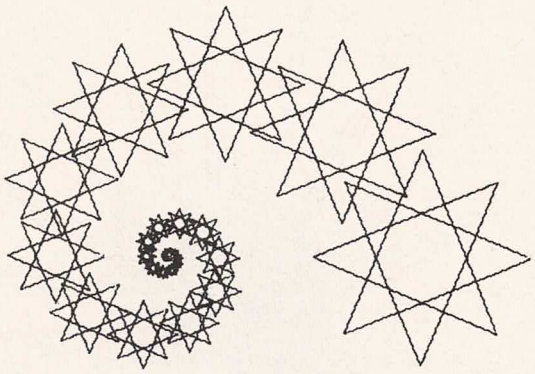
Modifications pour le dessin 25

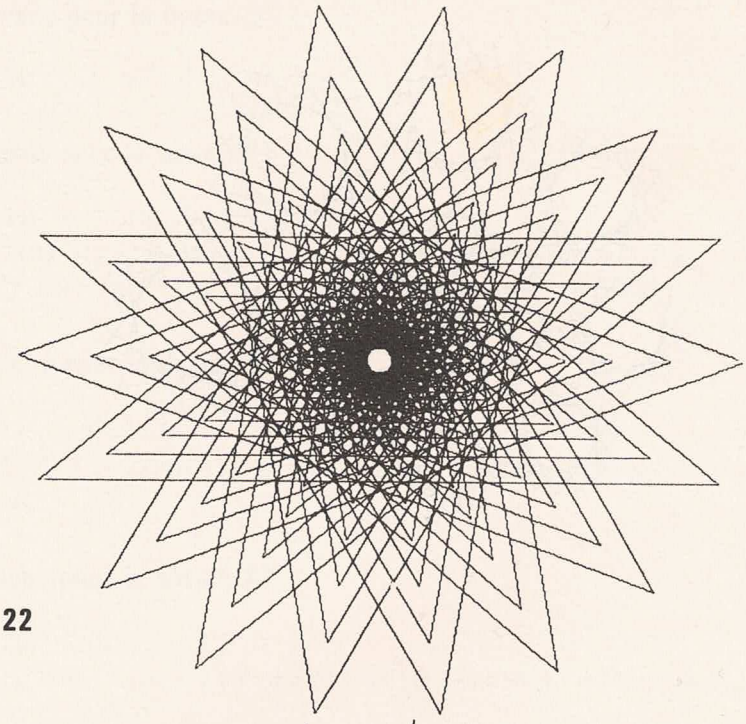
```
10 'COMPOSITION 2
100 K1=20:N=60:K=8:H=1:R1=NP*.05:R=NP*.4
5:RR=.945
```



20

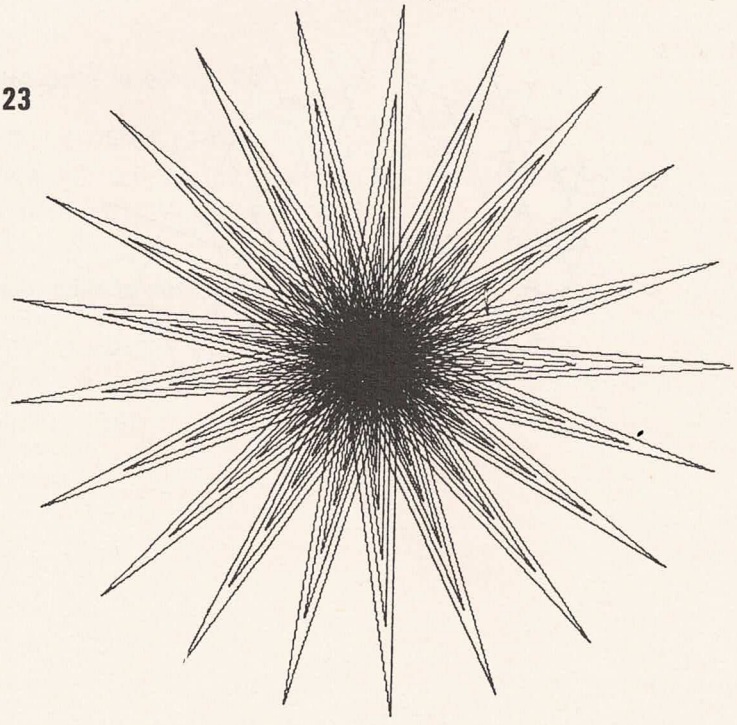
21

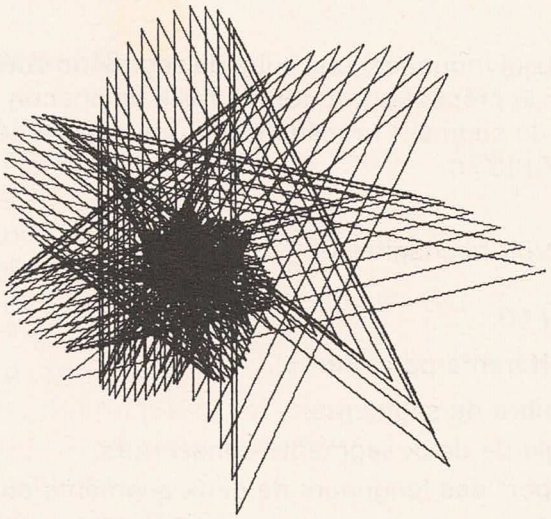




22

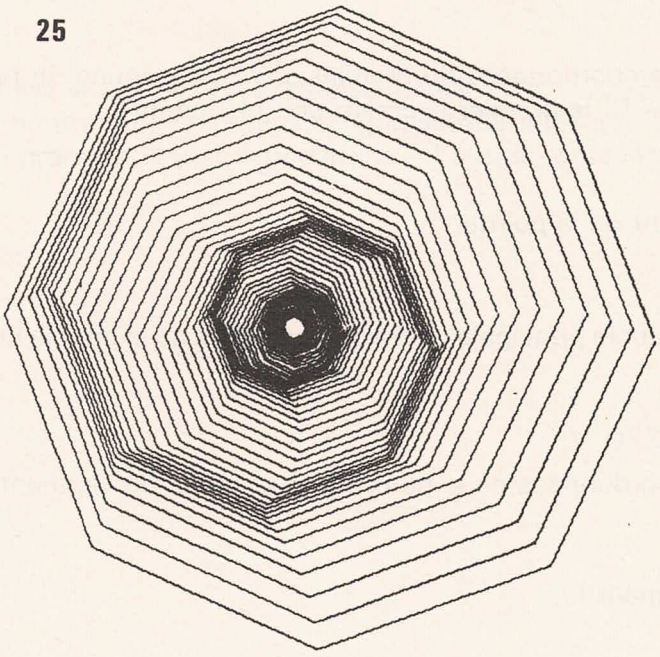
23





24

25



Le programme JOLIGONES

Un joligone (ou jolygone) est une suite de segments dont chacun fait un angle AN, fixe, avec le précédent, et dont la taille de chacun est obtenue en multipliant la taille du segment précédent par une constante RA (voir les références [2] [9] [36] [37]).

Ligne 50

Initialisation. Voir le chapitre 0.

Lignes 100, 110

On fixe les différents paramètres.

K : nombre de segments,
AN : angle de deux segments consécutifs,
RA : rapport des longueurs de deux segments consécutifs,
AA : angle du premier segment avec l'horizontale,
RR : longueur du premier segment.

Ligne 120

On calcule les coordonnées du premier point du joligone de façon à ce que (lorsque AA = 0) le premier segment soit bien centré.

Ligne 130

Positionnement de la pointe.

Ligne 200

Début de la boucle principale ; à chaque parcours de la boucle un segment est dessiné.

Lignes 210, 220

Calcul des coordonnées de la deuxième extrémité du segment I.

Ligne 230

Tracé du segment I.

Ligne 240

Calcul de l'angle du segment I + 1 avec l'horizontale et de la longueur du segment I + 1.

Lignes 300, 500

Fin de la boucle principale. Fin du programme.

Programme de base « JOLIGONES »

```
10 ' JOLIGONES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 K=200:AN=15*PI/31:RA=.98
110 AA=0:RR=.80*NP
120 X=(NP-RR)/2:Y=0:X%=INT(X):Y%=INT(Y)
130 LPRINT"M";X%;",";Y%
200 FOR I=0 TO K
210 X=X+RR*COS(AA):X%=INT(X)
220 Y=Y+RR*SIN(AA):Y%=INT(Y)
230 LPRINT"D";X%;",";Y%
240 AA=AA+AN:RR=RR*RA
300 NEXT I
500 END
```

Les différents essais donnés en exemples (dessins 26, 27, 28, 29, 30, 31, 32, 33) vous montrent de nombreuses figures différentes que l'on peut obtenir avec ce programme. Il y en a une infinité d'autres, essayez-en quelques-unes !

Modifications pour le dessin 26

```
10 'JOLIGONES
100 K=200:AN=15*PI/31:RA=.98
```

Modifications pour le dessin 27

```
10 ' JOLIGONES
100 K=120:AN=29*PI/30:RA=.98
```

Modifications pour le dessin 28

```
10 ' JOLIGONES
100 K=200:AN=PI/4:RA=.98
110 AA=0:RR=.40*NP
```

Modifications pour le dessin 29

```
10 'JOLIGONES
100 K=2000:AN=PI/20:RA=.998
110 AA=0:RR=.065*NP
```

Modifications pour le dessin 30

```
10 'JOLIGONES
100 K=200:AN=4*PI/5+.02:RA=.99
```

Modifications pour le dessin 31

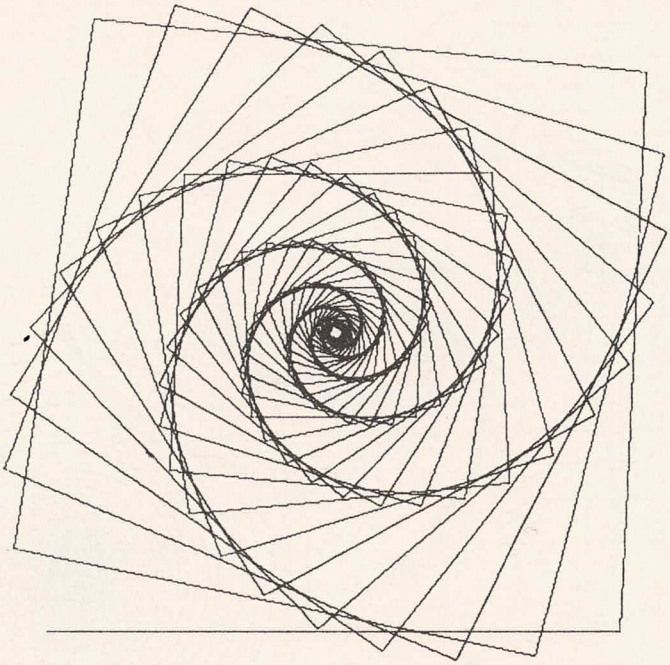
```
10 'JOLIGONES
100 K=100:AN=6*PI/7:RA=.98
```

Modifications pour le dessin 32

```
10 'JOLIGONES
100 K=300:AN=2*PI/5+.01:RA=.993
110 AA=0:RR=.60*NP
```

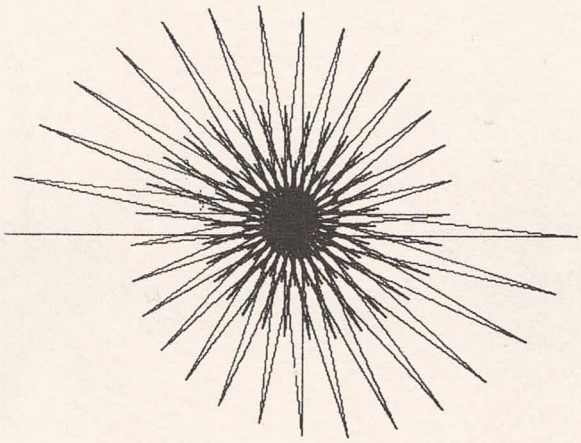
Modifications pour le dessin 33

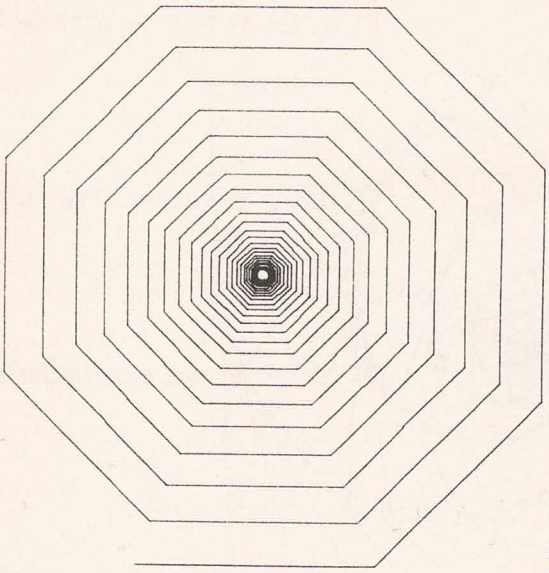
```
10 'JOLIGONES
100 K=400:AN=19*PI/60:RA=.996
110 AA=0:RR=.40*NP
220 Y=Y+RR*SIN(AA):Y%=INT(1.7*Y)
```



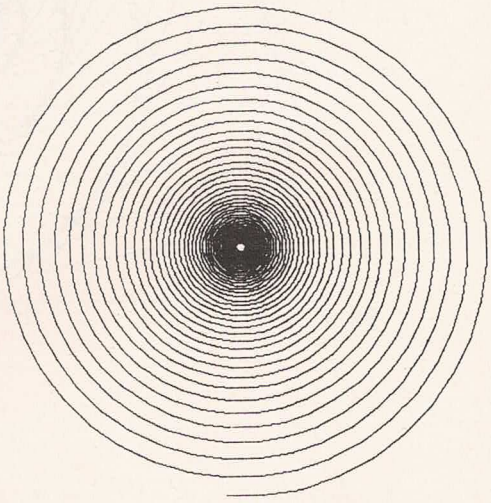
26

27



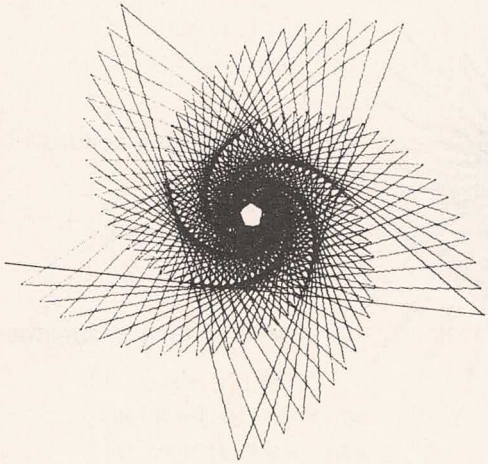


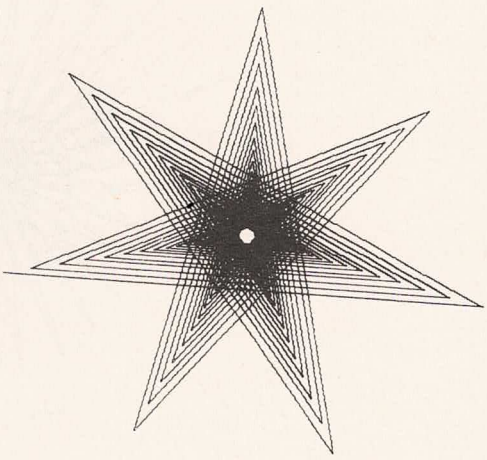
28



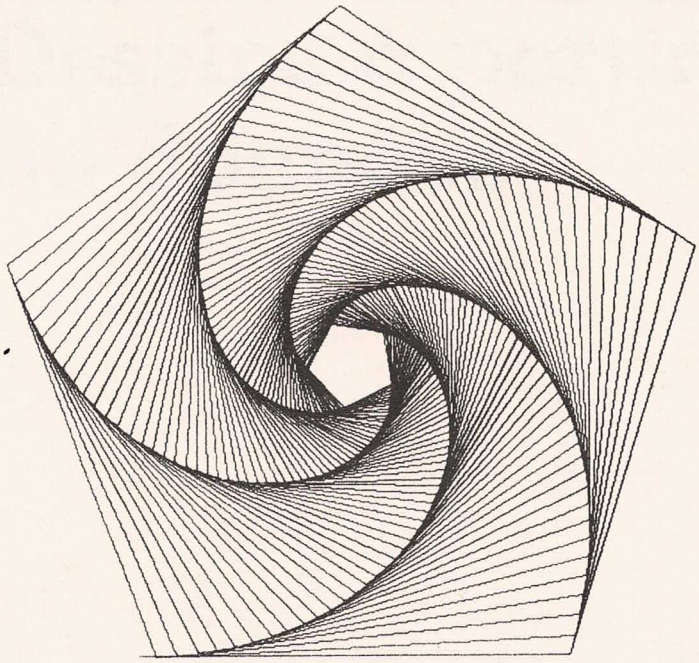
29

30



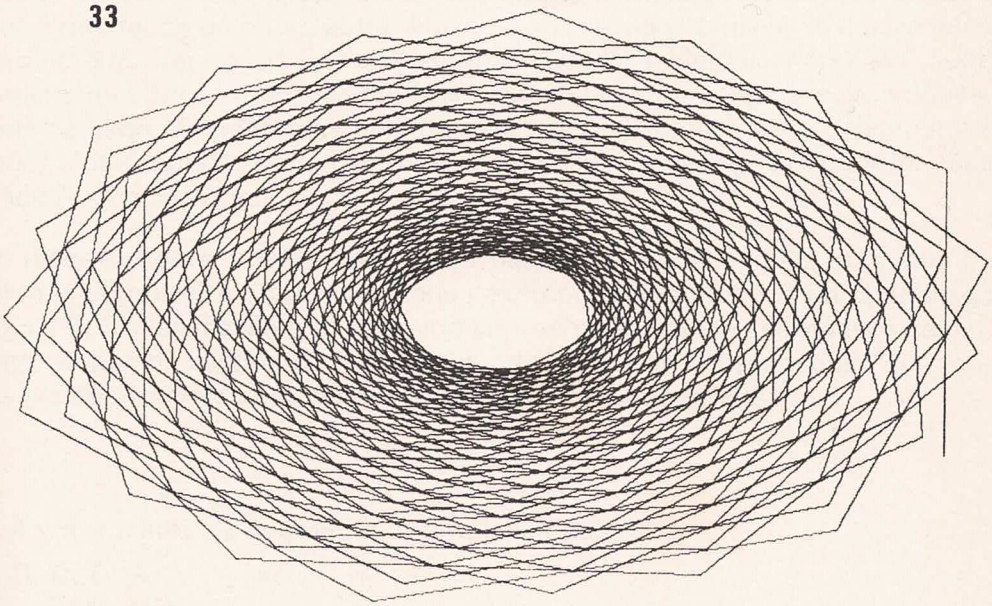


31



32

33



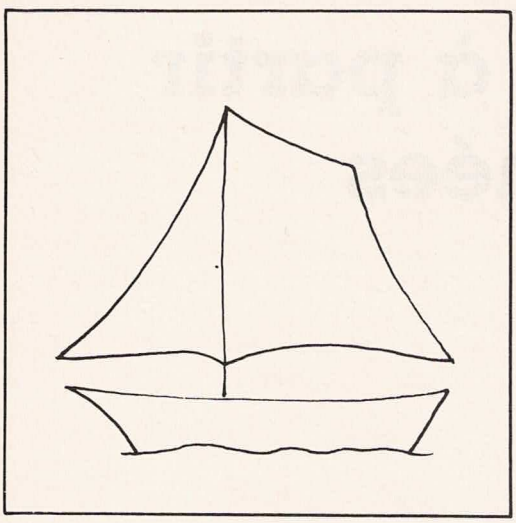
2 Dessins à partir de données

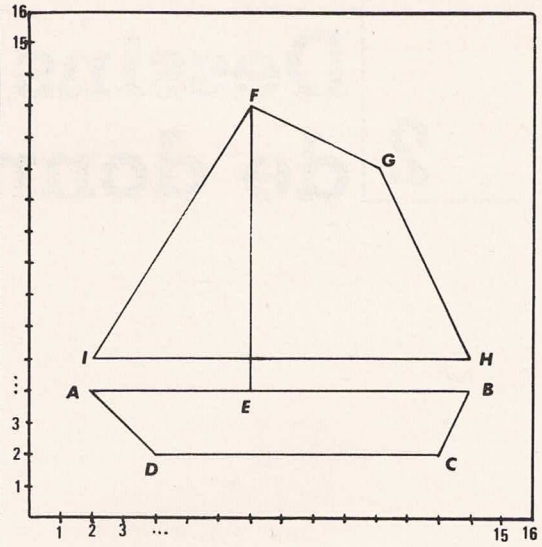
Lorsqu'on veut réaliser un dessin non mathématique avec un ordinateur, il faut d'une façon ou d'une autre, entrer la description détaillée du dessin dans l'ordinateur. Ceci peut se faire avec une table à digitaliser, ou avec une caméra numérique ou par d'autres moyens technologiques sophistiqués. Mais ce genre de méthodes sort du cadre de ce livre et nous n'expliquerons que la plus simple de toutes les méthodes, celle du repérage des coordonnées grâce à un quadrillage.

On place la figure que l'on veut saisir dans un quadrillage plus ou moins fin selon la complexité du dessin. Dans l'exemple, nous prenons un quadrillage 16 x 16. On réalise alors une version approchée du dessin, composée de plusieurs suites de segments. Il est commode (mais non indispensable) que toutes les coordonnées des points soient des nombres entiers.

Ici il y a 2 suites de segments déterminées par :

A, B, C, D, A et
E, F, G, H, I, F





Ce qui donne deux suites de coordonnées

(2 , 4) , (14 , 4) , (13 , 2) , (4 , 2) , (2 , 4) et
(7 , 4) , (7 , 13) , (11 , 11) , (14 , 5) , (2 , 5) , (7 , 13)

Dans nos programmes, nous coderons ces données en ponctuant avec 1000 pour indiquer la fin d'une suite de segments, et avec 2000 pour indiquer la fin du dessin. Pour le bateau, ceci donne :

2 , 4 , 14 , 4 , 13 , 2 , 4 , 2 , 2 , 4 , 1000 , 7 , 4 , 7 , 13 , 11 , 11 , 14 , 5 , 2 , 5 , 7 , 13 , 2000.

Ces données seront placées dans les lignes « data » (ou dans des fichiers) que le programme ira lire pour réaliser le dessin.

Si le but était simplement de reproduire la figure saisie sans rien en faire de particulier, ce genre de travail serait pénible et de peu d'intérêt. C'est seulement par l'utilisation répétée et variée de la figure saisie que se justifie la longue phase peu amusante de la mise au point des données.

Le programme « CHEVAL »

Nous allons illustrer le principe des coordonnées avec un motif plus compliqué que le motif du bateau qui a servi à donner les explications générales.

Nous avons utilisé un quadrillage de 40 sur 40 (au lieu de 16 sur 16 pour le bateau) et codé un motif de cheval ailé d'après une œuvre du graveur hollandais Escher ([11] [21] [22] [31]). Ce motif possède une propriété extraordinaire : lorsqu'on le répète, on obtient différentes figures qui s'emboîtent exactement les unes dans les autres sans recouvrement, ni interstice (voir le dessin 39).

Ligne 50
Initialisation. Voir le chapitre 0.

Lignes 1000 à 1500

Données correspondant au cheval codé selon le principe expliqué en introduction.

Lignes 2000 à 2300

Boucle de lecture des données et de tracé.

A chaque parcours de cette boucle, un nouveau point est lu et le segment correspondant est dessiné. La variable B1 est une variable booléenne qui indique qu'on doit se déplacer pointe levée (B1 = 0) ou pointe baissée (B1 = 1).

Lignes 2200 et 2210

Opérations qui traduisent les coordonnées de la grille 40 x 40 en coordonnées de la grille NP x NP.

Ligne 3000

Fin du programme.

Programme de base « CHEVAL »

```
10 'CHEVAL
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
1000 DATA 1000, 10,10, 8,12, 9,16, 12,17
, 13,18, 14,20
1010 DATA 1000, 13,18, 12,19, 9,21, 9,20
, 10,19, 9,17, 7,20, 8,22, 12,22
1020 DATA 1000, 12,20, 12,22, 13,26, 16,
31, 18,31, 19,32
1030 DATA 1000, 16,31, 14,31, 14,32
1040 DATA 1000, 14,31, 10,30, 12,31, 10,
32, 10,34, 11,34, 11,33, 10,33
1050 DATA 1000, 12,32, 13,31
1060 DATA 1000,10,34, 16,36
1070 DATA 1000,16,35, 16,37, 18,35, 17,3
4
1080 DATA 1000, 17,36, 20,36, 22,32, 19,
26
1090 DATA 1000, 20,36, 22,36, 22,34, 24,
32, 24,30, 19,26, 18,23, 21,22, 21,24
1100 DATA 30,30, 34,31, 36,31, 33,26, 32
,22, 28,22, 27,20, 29,17, 30,19, 29,20
1110 DATA 29,21, 32,19, 33,18, 32,17, 29
,16, 28,12, 30,10, 21,4, 21,2
1120 DATA 18,3, 19,6, 24,10, 24,12, 22,1
4, 22,16, 23,17
1130 DATA 1000, 22,16, 17,16, 16,17, 17,
18
1140 DATA 1000, 16,17, 16,16, 10,14, 10,
12, 12,11, 10,10
1150 DATA 1000, 21,21, 22,24, 30,30
1160 DATA 1000, 24,24, 34,28
1170 DATA 1000, 25,23, 33,26
1190 DATA 1000, 25,21, 27,20
1200 DATA 1000, 23,21, 24,19
1210 DATA 1000, 27,20, 22,19, 22,21
1220 DATA 1000, 22,19, 21,20
1230 DATA 1000, 13,34, 15,35, 16,34, 16,
33
```

```
1240 DATA 1000, 15,35, 15,34, 16,34, 15,
34, 15,35
1250 DATA 1000, 24,12, 26,10, 19,5, 19,3
1260 DATA 1000, 28,22, 25,22
1500 DATA 2000
2000 '
2100 READ A
2110 IF A=2000 THEN 2400
2120 IF A=1000 THEN B1=0 :READ A
2130 READ B
2200 X%=NP*A/40
2210 Y%=NP*B/40
2300 IF B1=0 THEN B1=1 :LPRINT"M";X%;",";
Y%
2310 IF B1=1 THEN LPRINT"D";X%;",";Y%
2320 GOTO 2100
2400 REM
3000 END
```

Pour le dessin 34, on s'est contenté de reproduire le motif une seule fois.

Les dessins 35, 36, 37, 38, 39, 40, 41, 42 et 43 vous montrent des exemples d'utilisation de ce motif.

Dessin 35

Ligne 2000

Début de la boucle principale. A chaque parcours de la boucle un cheval est dessiné.

Ligne 2010

On fixe l'angle avec lequel on va faire tourner le motif.

Ligne 2200

On ramène le dessin à un format plus petit.

Ligne 2210

On fait tourner le motif de l'angle AN.

Les formules générales qui permettent de faire tourner le point de coordonnées X, Y autour du point O de coordonnées OX, OY d'un angle A sont :

XX = OX + COS (A) * (X — OX) — SIN (A) * (Y — OY)
YY = OY + SIN (A) * (X — OX) + COS (A) * (Y — OY)

Dessin 36

Lignes 2200 - 2210

Après avoir rapetissé le dessin (A/80, B/80), on le diminue une seconde fois d'un facteur .5 ^ I : (A/80 * [.5 ^ I], B/80 * [.5 ^ I]) ; ensuite lorsque J = 1, on fait une symétrie par rapport à l'axe OX ([1 — 2 * J] * A/80 * [.5 ^ I], B/80 * [.5 ^ I]) ; ensuite on déplace le dessin de .5 dans le sens des X et de 1 — (.5) ^ I dans le sens des Y, (.5 + [1 — 2 * J]) * A/80 * (.5 ^ I), 1 — [.5 ^ I]) + B/80 * [.5 ^ I]) ; ensuite on ramène le tout à la taille NP x NP d'où la formule pour X %, Y %.

En faisant varier I et J, on obtient ces chevaux qui se regardent et diminuent vers l'infini.

Les dessins 37 et 38 sont basés sur des idées analogues. Les dessins 39 et 40 exploitent la possibilité d'emboîter les différents chevaux quand on les translate de I * 20 ou J * 20 (dans le cadre 40 x 40 qui a servi au repérage des coordonnées).

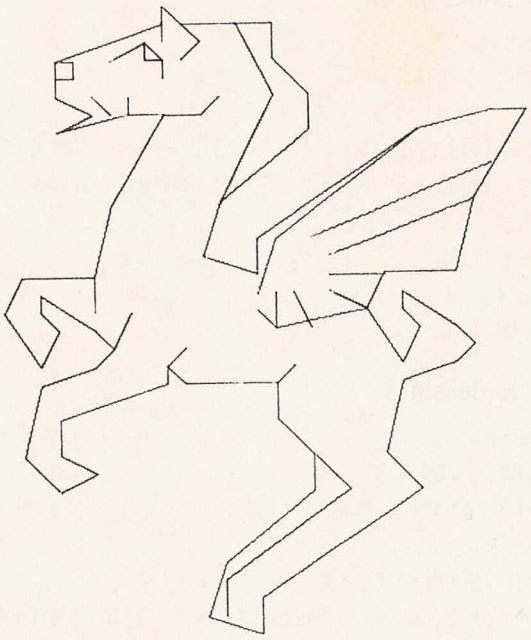
Les dessins 41, 42 et 43 sont construits à partir de pavages analogues à ceux des dessins 39 et 40, auxquels on fait subir des déformations (lignes 2200, 2210 pour les dessins 41 et 43 lignes 2220-2260 pour le dessin 42). Le principe de ces déformations sera réutilisé et expliqué plus en détail aux chapitres 7 et 8.

Modifications pour le dessin 34

```
10 'CHEVAL
```

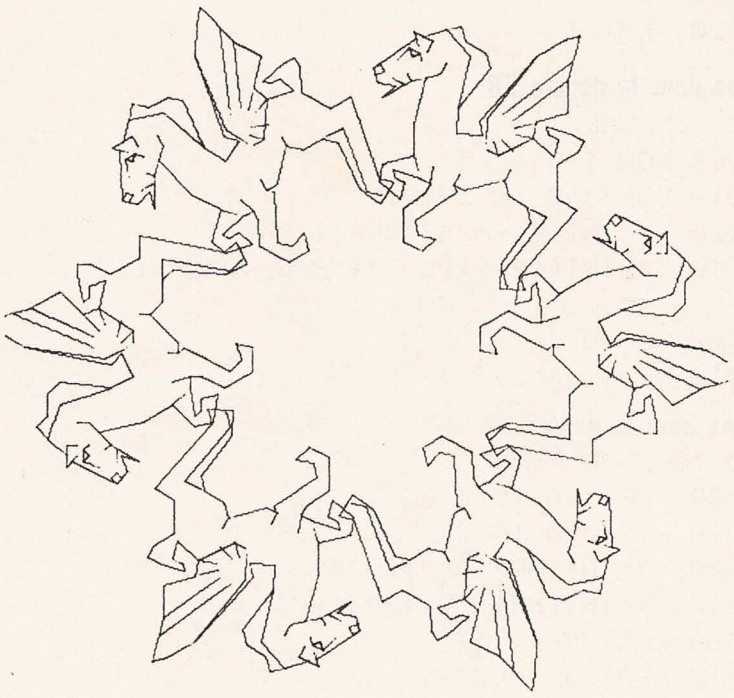
Modifications pour le dessin 35

```
10 'CHEVAL
2000 FOR I=0 TO 5
2010 AN=2*I*PI/6+PI/12:CO=COS(AN):SI=SIN
(AN)
2200 X=A/90:Y=B/90
2210 X%=INT(NP*(.5+CO*X-SI*Y)):Y%=INT(NP
*(.5+SI*X+CO*Y))
2400 REM
2410 RESTORE
2420 NEXT I
```



34

35



Modifications pour le dessin 36

```
10 'CHEVAL
2000 FOR I=0 TO 5
2020 FOR J=0 TO 1
2200 X%=INT(NP/2+(1-2*J)*NP*A/80*(.5^I))
2210 Y%=INT(NP-NP*(.5)^I+NP*B/80*(.5^I))
2400 REM
2410 RESTORE
2420 NEXT J
2430 NEXT I
```

Modifications pour le dessin 37

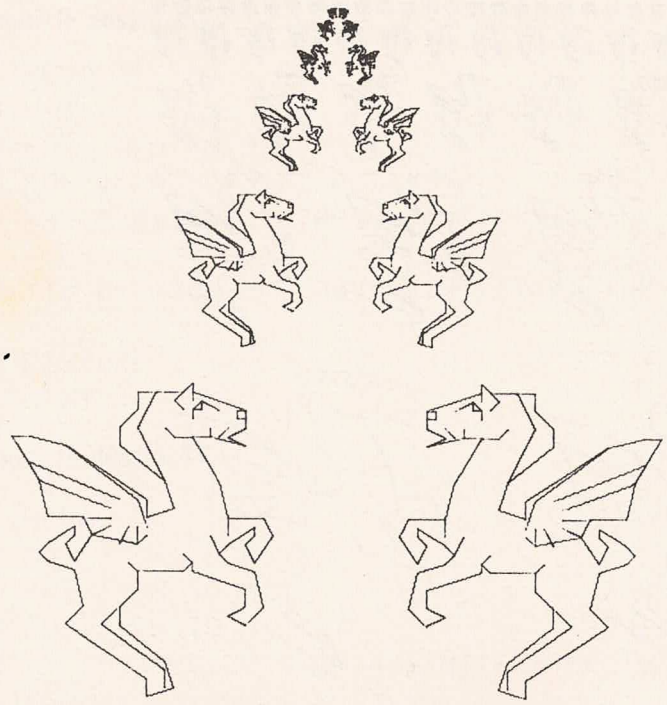
```
10 'CHEVAL
2000 FOR I=0 TO 15
2010 AN=2*I*PI/6+PI/12:CO=COS(AN):SI=SIN
(AN):R=.87^I
2200 X=.15+A/110:Y=.15+B/110
2210 X%=INT(NP*(.5+R*(CO*X-SI*Y))):Y%=IN
T(NP*(.5+R*(SI*X+CO*Y)))
2400 REM
2410 RESTORE
2420 NEXT I
```

Modifications pour le dessin 38

```
10 'CHEVAL
2000 FOR I=0 TO 5
2010 FOR J=0 TO 2^I-1
2200 X%=INT((J+A/40)*NP*(.5^I))
2210 Y%=INT((2-2*.5^I)*NP+B/40*NP*(.5^I)
)
2400 RESTORE
2410 NEXT J:NEXT I
```

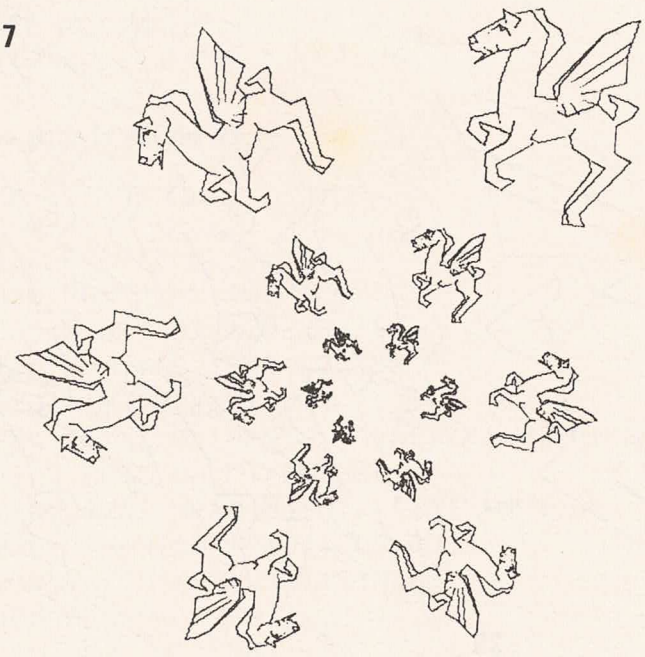
Modifications pour le dessin 39

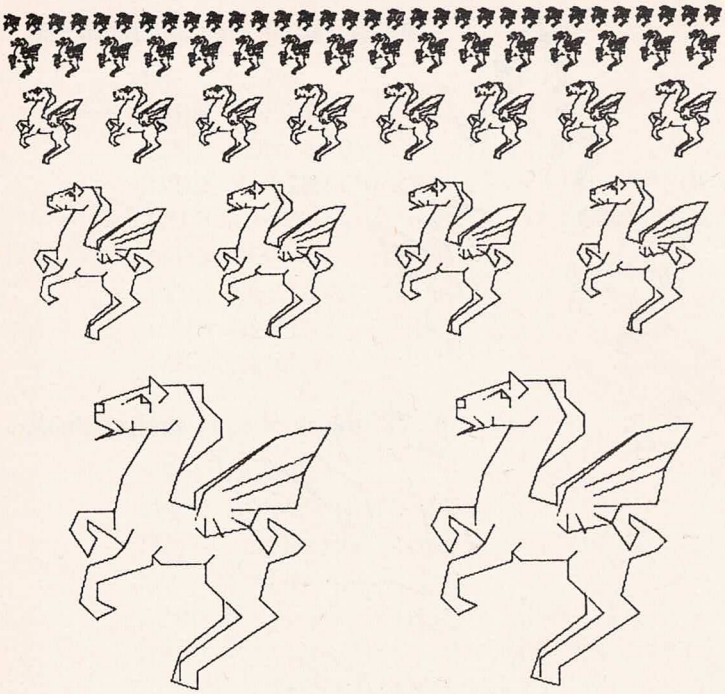
```
10 'CHEVAL
2000 FOR I=0 TO 2
2010 FOR J=0 TO 2
2200 X%=INT(NP*(A+J*20)/80)
2210 Y%=INT(NP*(B+I*20)/80)
2400 RESTORE
2410 NEXT J :NEXT I
```

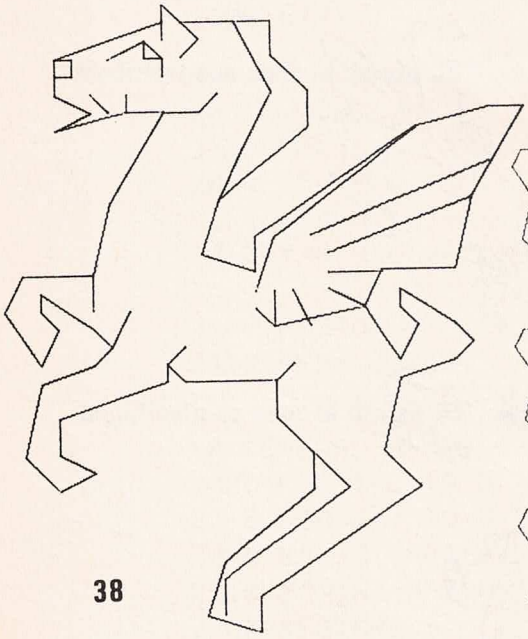


36

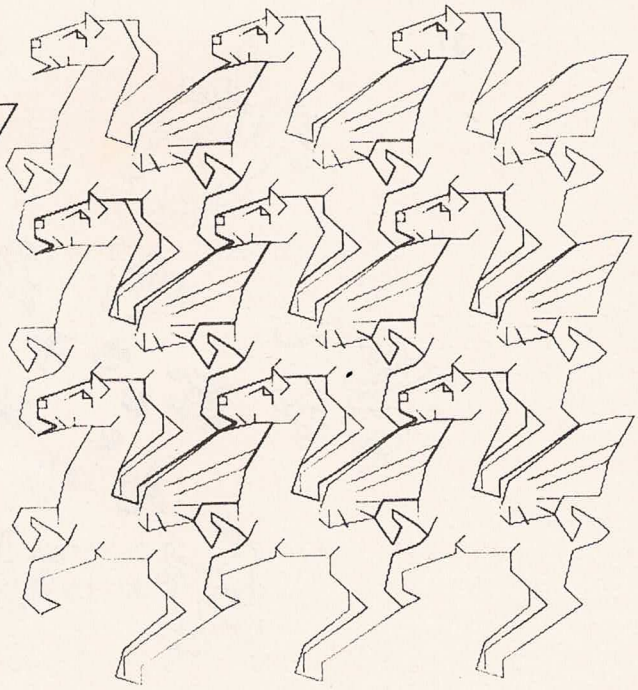
37







38



39

Modifications pour le dessin 40

```
10 'CHEVAL
2000 N=4
2010 FOR I=-N TO N
2020 FOR J=-ABS(I) TO ABS(I)
2200 XX=(A+J*20-20)/100:X%=INT((XX+1)*NP
/2)
2210 YY=(B+I*20-20)/100:Y%=INT((YY+1)*NP
/2)
2400 RESTORE
2410 NEXT J :NEXT I
```

Modifications pour le dessin 41

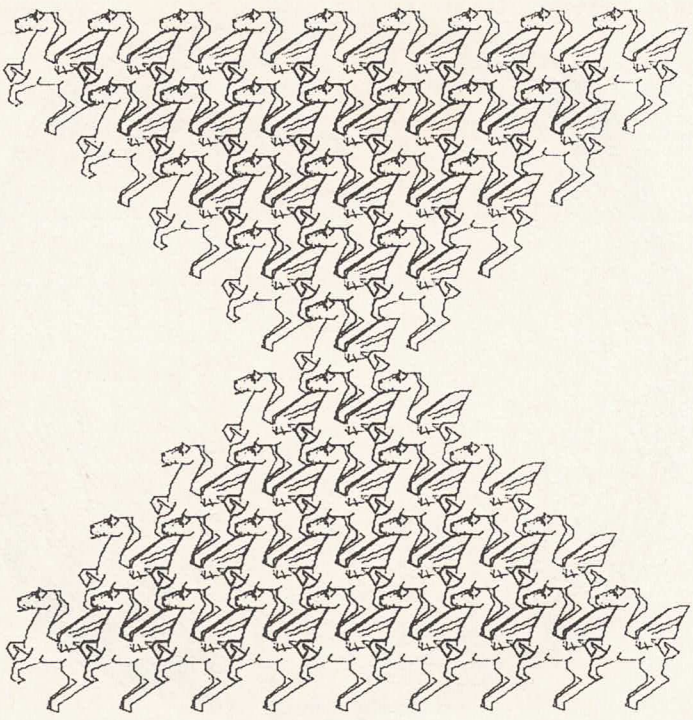
```
10 'CHEVAL
2000 N=4
2010 FOR I=-N TO N
2020 FOR J=-N TO N
2200 XX=(A+J*20-20)/100:X%=INT((XX*ABS(X
X)+1)*NP/2)
2210 YY=(B+I*20-20)/100:Y%=INT((YY*ABS(Y
Y)+1)*NP/2)
2400 RESTORE
2410 NEXT J :NEXT I
```

Modifications pour le dessin 42

```
10 'CHEVAL
2000 N=4
2010 FOR I=-N TO N
2020 FOR J=-N TO N
2200 X=NP*(A+J*20-20)/80
2210 Y=NP*(B+I*20-20)/80
2220 DI=SQR(X*X+Y*Y)
2230 IF X<>0 THEN AN=ATN(Y/X)+PI*(1-SGN(
X))/2 ELSE AN=PI/2*SGN(Y)
2240 DI=DI/NP*3:DI=DI/(1+DI)*NP*.65
2250 X%=INT(NP/2+DI*COS(AN))
2260 Y%=INT(NP/2+DI*SIN(AN))
2400 RESTORE
2410 NEXT J :NEXT I
```

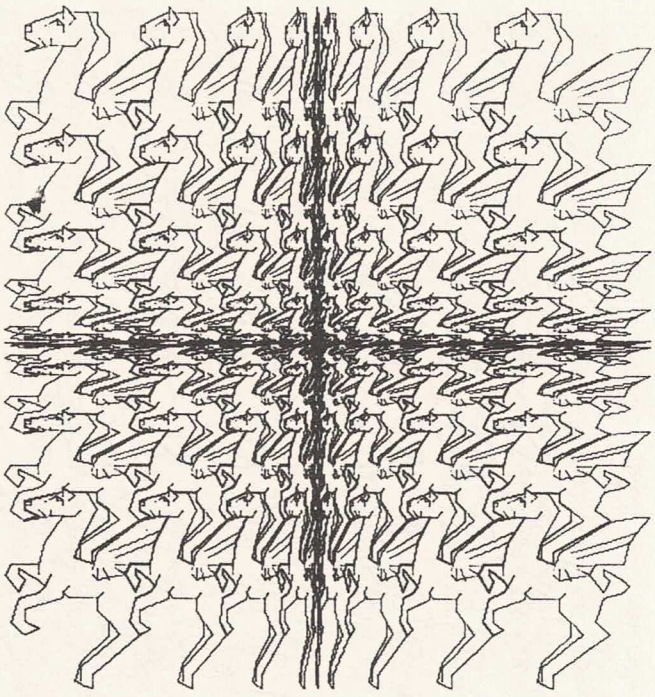
Modifications pour le dessin 43

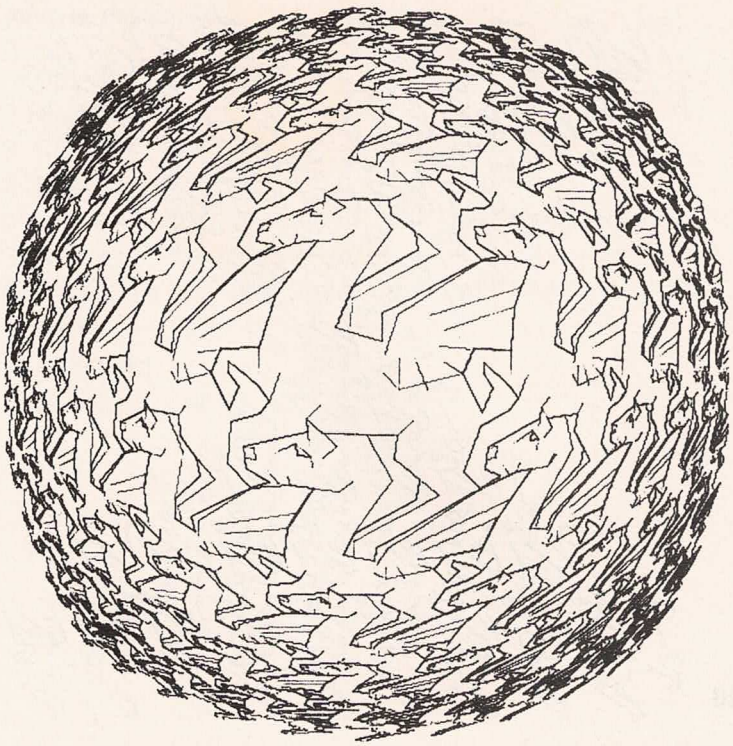
```
10 'CHEVAL
2000 N=4
2010 FOR I=-N TO N
2020 FOR J=-N TO N
2200 XX=(A+J*20-20)/100:X%=INT((ABS(XX)^
.7*SGN(XX)+1)*NP/2)
2210 YY=(B+I*20-20)/100:Y%=INT((ABS(YY)^
.7*SGN(YY)+1)*NP/2)
2400 RESTORE
2410 NEXT J :NEXT I
```



40

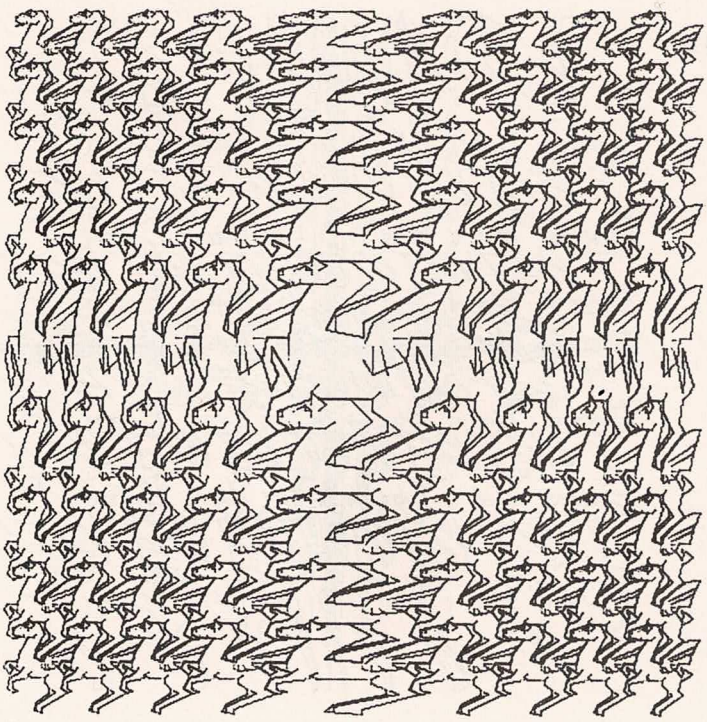
41





42

43



Programmes LION, OISEAUX-POISSONS, SMURF

Les programmes « LION », « OISEAUX-POISSONS », « SMURF » sont écrits exactement sur le modèle du programme « CHEVAL » en utilisant d'autres motifs.

Les motifs du lion et des oiseaux-poissons sont dus aussi à Escher. Le motif du schtroumpf (qui se dit smurf en anglais !) est dû à Peyo. Le repérage soigné des quatre figures de base de ce chapitre a été réalisé par Martine Delahaye.

Programme de base « LION »

```
10 'LION
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
1000 DATA 1000,-2.5,0, -2,1, -1,2, 0,7,
1,7, 2,8, 2,11, 3,14, 3.5,13.5
1010 DATA 2.5,11, 2.5,9
1020 DATA 1000,3.5,13.5, 4,13, 3,11, 3,9
, 3,11, 4,13, 5,12, 3.5,11, 3.5,9
1030 DATA 3.5,11, 5,12, 5,11, 4,10, 4,9,
 8,9, 7,11,  8,13, 10,14, 12,13, 13,11
1040 DATA 12,11,11,10, 12,8, 13,7, 14,2,
 15,2, 16,1, 16,0, 12,0, 12,2, 11,5
1050 DATA 11.5,6, 11,5, 9,3, 9,2, 10,1,
10,0, 6,0, 7,2, 8,6, 7,2, 6,4,  4,5
1060 DATA 5,7, 4,8,5,7, 4,5, 2,4, 1,2, 2
,2, 3,1, 2.5,0,-2.5,0
1070 DATA 1000, 6,4, 7.5,3.5
1080 DATA 1000, 12,11, 10,10.5, 9,10.5
1090 DATA 1000, 12.5,12, 12,12, 11,11.5,
 12,12, 12,12.5, 11.5,12.5, 10.5,13
1100 DATA 10,13, 10,13.5, 10.5,13.5, 10.
5,13, 11.5,12.5,  12,12.5,  12,13
1110 DATA 1000,7.5,12, 8.5,12, 8.5,11.5
1200 DATA 2000
2000 '
2100 READ A
2110 IF A=2000 THEN 2400
2120 IF A=1000 THEN B1=0 :READ A
2130 READ B
```

```
2200 X%=INT(NP*(A+5)/25)
2210 Y%=INT(NP*(B+5)/25)
2300 IF B1=0 THEN B1=1:LPRINT"M";X%;",";
Y%
2310 IF B1=1 THEN LPRINT"D";X%;",";Y%
2320 GOTO 2100
2400 REM
3000 END
```

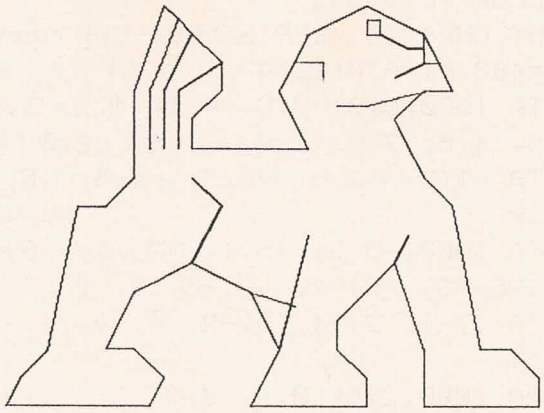
Modifications pour le dessin 44

```
10 'LION
```

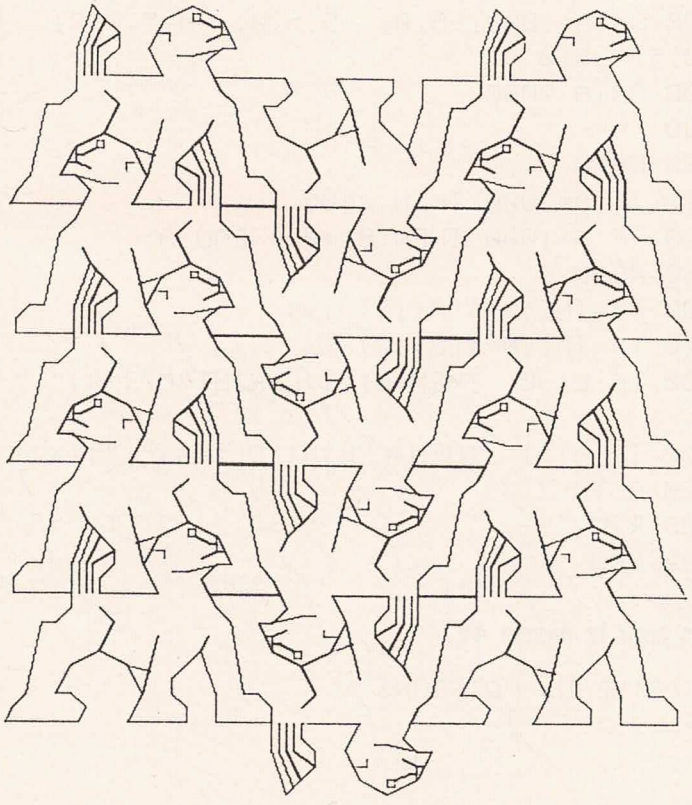
Modifications pour le dessin 45

```
10 'LION
2000 FOR I=0 TO 4
2010 FOR J=0 TO 2
2200 X%=INT(NP*(7+(1-2*(I MOD 2))*(7-A)+
4+14*J)/50)
2210 Y%=INT(NP*(4.5+(1-2*(J MOD 2))*(4.5
-B)+4+9*I)/50)
2400 RESTORE
2410 NEXT J:NEXT I
```

44



45



Programme de base « OISEAUX - POISSONS »

```
10 'OISEAUX-POISSONS
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
1000 DATA 1000,0,0, 2,0, 4,1, 4,2, 3,2,
2,3, 4,5, 4,6, 2,5, 2,6, -1,5, -2,3
1010 DATA -1,2, -2,2, -3,3, -4,3, -5,2,
-4,2, 0,0
1020 DATA 1000,-5,2, -5,1, -7,-1, -6,-2,
 -5,-2, -5,-3, -2,-2, -2,-3, 0,-2
1030 DATA 1,-1, 2,-1, 3,-2, 4,-2, 3,-1,
4,1
1040 DATA 1000,2,5, 0,4, 0,2
1050 DATA 1000,-2,1, -5,1, -4,-1, -3,0,
-3,-1, -4,-1, -5,-2, 0,-2
1060 DATA 1000, -7,-1, -6,-1
1070 DATA 1000,-4,2.5, -4,2.8, -4.3,2.8,
 -4.3,2.5, -4,2.5
1080 DATA 1000,-5,0, -5.5,0, -5.5,0.5,
5,0.5, -5,0
1200 DATA 2000
2000 '
2100 READ A
2110 IF A=2000 THEN 2400
2120 IF A=1000 THEN B1=0 :READ A
2130 READ B
2200 X%=INT(NP*(A+10)/15)
2210 Y%=INT(NP*(B+10)/15)
2300 IF B1=0  THEN B1=1:LPRINT"M";X%;","
;Y%
2310 IF B1=1  THEN LPRINT"D";X%;",";Y%
2320 GOTO 2100
2400 REM
3000 END
```

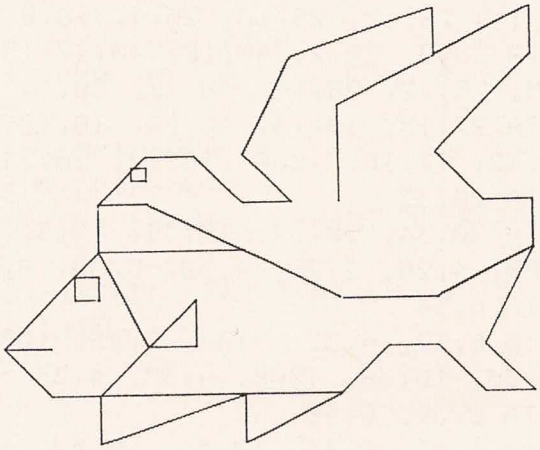
Modifications pour le dessin 46

```
10 'OISEAUX-POISSONS
```

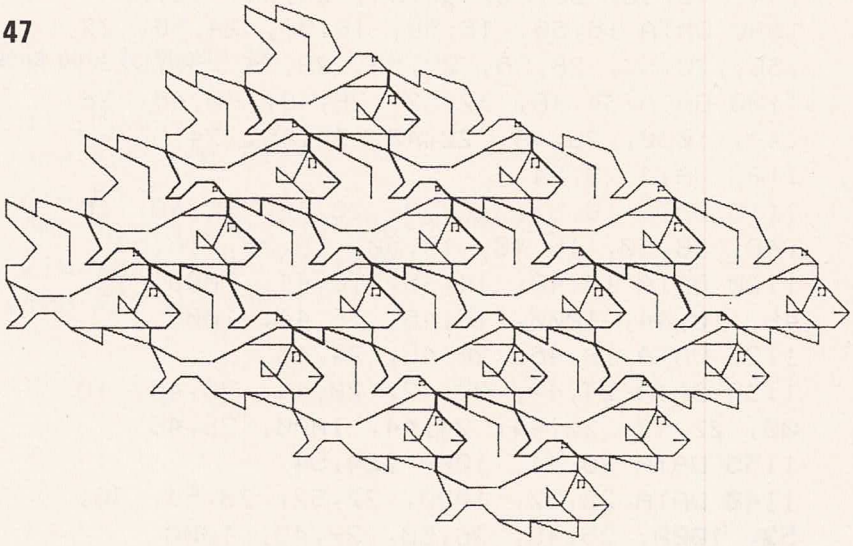
Modifications pour le dessin 47

```
10 'OISEAUX-POISSONS
2000 FOR I=1 TO 4
2010 FOR J=1 TO 4
2200 X%=INT(NP*(B+4*I+4*J)/45)
2210 Y%=INT(NP*(A+15-5*I+9*J)/45)
2400 RESTORE
2410 NEXT J:NEXT I
```

46



47



Programme de base « SMURF »

```
10 'SMURF
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
1000 DATA 1000
1010 DATA 12,12, 14,8, 14,4, 12,2, 8,2,
4,4, 0,10, 0,20, 4,26, 6,28, 12,28
1020 DATA 14,26, 14,22, 12,16, 12,12, 20
,14, 24,14, 28,12, 28,10, 26,4, 28,0
1030 DATA 36,0, 38,2, 40,10, 40,22, 36,2
6, 28,26, 26,22, 28,14, 28,12, 28,14
1040 DATA 27,18, 18,18, 16,20, 16,18, 20
,14, 16,18, 12,16, 1000, 16,20, 16,24
1045 DATA 20,32
1050 DATA 20,34, 20,32, 12,34, 12,32, 10
,28, 1000, 4,26, 2,28, 4,30, 8,30, 6,32
1055 DATA 6,34
1060 DATA 6,32, 4,32, 2,30, 2,28, 1000,
8,30, 8,36, 10,38, 1000, 4,32, 4,34
1065 DATA 8,38, 6,40
1070 DATA 6,42, 8,44, 10,44, 10,42, 12,4
2, 12,38, 16,36, 32,36, 38,40, 40,44
1080 DATA 38,42, 36,46, 30,48, 36,48, 40
,44, 40,56, 36,62, 32,64, 24,64, 18,62
1090 DATA 16,60, 16,58, 18,56, 24,56, 22
,56, 20,53, 28,56, 22,54, 28,54, 32,52
1100 DATA 34,48, 32,52, 28,48, 30,46, 28
,44, 1000, 28,48, 22,48, 24,48, 24,52
1105 DATA 22,54
1110 DATA 18,52, 18,50, 20,48, 12,48, 16
,48, 18,50, 16,48, 16,50, 18,52, 16,50
1120 DATA 16,48, 14,46, 16,44, 1000, 12,
48, 10,44, 1000, 16,46, 18,44, 1000
1125 DATA 18,46, 26,46, 24,46
1130 DATA 24,44, 22,42, 20,44, 20,46, 10
00, 22,42, 22,44, 24,44, 1000, 28,46
1135 DATA 26,44, 1000 ,24,54
1140 DATA 25,52, 1000, 27,52, 28,54, 30,
52, 1000, 25,49, 26,50, 27,49, 1000
```

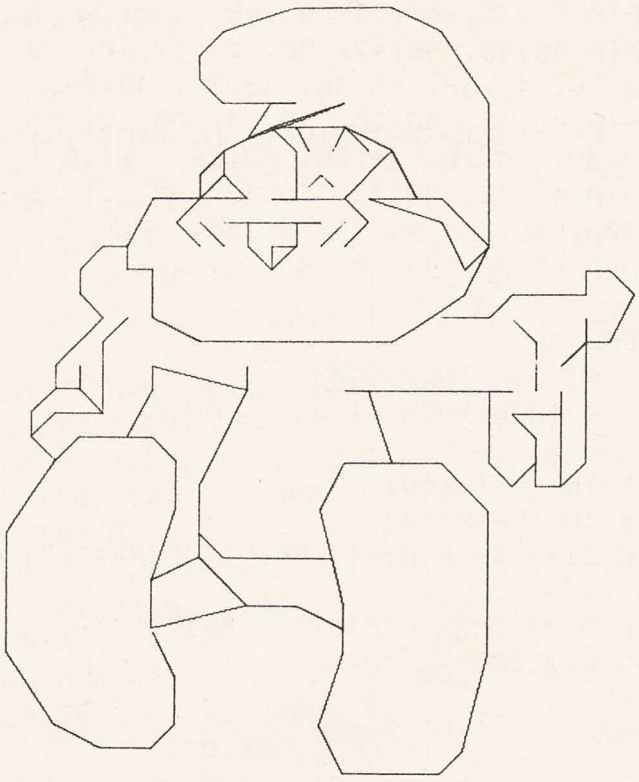
```
1145 DATA 36,38, 40,38, 42,40
1150 DATA 48,40, 48,42, 50,42, 52,40, 50
,36, 48,36, 48,38, 48,38, 48,36, 46,34
1160 DATA 48,36, 48,26, 46,24, 46,32, 46
,30, 42,30, 44,28, 44,26, 42,24, 40,26
1170 DATA 40,32, 42,32, 28,32, 30,32, 32
,26, 1000, 44,26, 44,24, 46,24, 1000
1175 DATA 42,38, 44,36, 44,32, 2000
2000 '
2100 READ A
2110 IF A=2000 THEN 2400
2120 IF A=1000 THEN B1=0 : READ A
2130 READ B
2200 X%=INT(NP*A/60)
2210 Y%=INT(NP*B/60)
2300 IF B1=0 THEN B1=1:LPRINT"M";X%;",";
Y%
2310 IF B1=1 THEN LPRINT"D";X%;",";Y%
2320 GOTO 2100
2400 REM
3000 END
```

Modifications pour le dessin 48

```
10 'SMURF
```

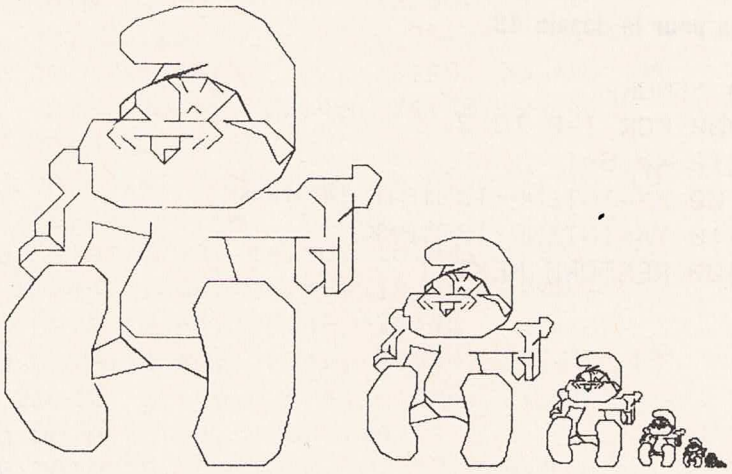
Modifications pour le dessin 49

```
10 'SMURF
2000 FOR I=0 TO 7
2010 K=.5^I
2200 X%=INT(NP/100*A*K+NP-NP*K)
2210 Y%=INT(NP/100*B*K)
2400 RESTORE:NEXT I
```



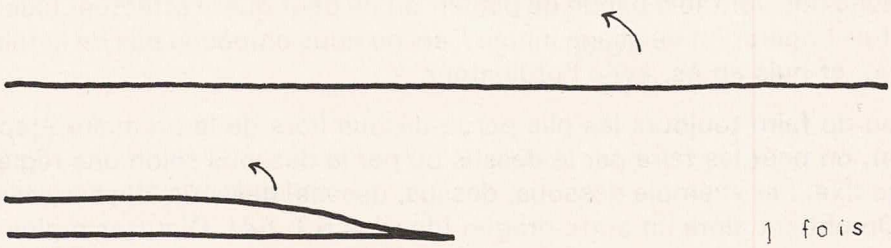
48

49

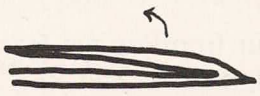


3 Dragons de papiers pliés

Imaginons que vous ayez une bande de papier entre les mains et que vous la pliiez 3 fois sur elle-même par le dessus :



1 fois



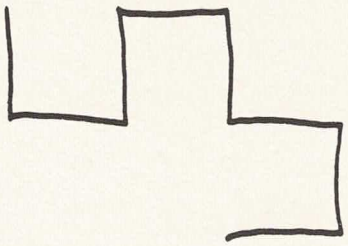
2 fois

3 fois

Maintenant vous la dépliez. Vous obtenez :

Il reste des plis marqués en bosses ou en creux.

Reprenez à nouveau la bande de papier et transformez chaque marque de pli en angle droit vers la gauche ou vers la droite selon le sens du pli. Vous obtenez :



Vous avez là un dragon de papiers pliés à l'ordre 3. Ce n'est pas extraordinaire !

Par contre, si au lieu de faire la première opération de pliage 3 fois, vous l'aviez fait 6 fois ce serait déjà plus intéressant (voir dessin 50). Et encore plus, si vous l'aviez fait 10 fois (dessin 51) ou même 14 fois (dessin 52). En fait, avec une véritable bande de papier, on ne peut guère effectuer plus de 7 ou 8 fois l'opération de pliage initial. Ceci ne nous empêche pas de la faire en pensée, et puis après, avec l'ordinateur.

Au lieu de faire toujours les plis par le dessus (lors de la première étape de pliage), on peut les faire par le dessus ou par le dessous selon une règle que l'on se fixe. Par exemple dessous, dessus, dessus, dessous, dessus, dessus, etc. On obtient alors un autre dragon (dessins 53, 54). D'autres règles donnent d'autres dragons encore (dessins 55 à 64). Tous sont extraordinaires sous plusieurs aspects :

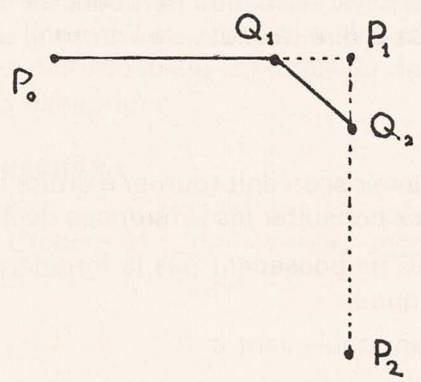
- jamais la courbe obtenue ne se recoupe elle-même ;
- ils ont des bords infiniment ciselés : on dit que leur frontière est fractale (nous verrons de nouveau des fractales dans les chapitres 4, 7 et 9) ;
- quand on dispose de plusieurs modèles d'un même dragon, on voit qu'ils s'emboîtent les uns dans les autres de façon parfaite (comme les motifs de Escher du chapitre 2).

Pour avoir plus de précisions sur les dragons de papiers pliés, consultez les références [3] [5] [15] [25] [32].

Le programme DRAGONS

Le programme qui suit exécute le dessin d'un dragon, à l'ordre que l'on veut et défini par la règle de pliage « dessus-dessous... » que l'on veut.

Pour permettre de bien suivre le tracé du dragon, nous avons arrondi les angles droits. Le principe utilisé est le suivant : on garde en mémoire dans les variables XØ, YØ, X1, Y1 les coordonnées des deux derniers points calculés PØ et P1, ensuite on calcule P2 et on joint les points Q1 et Q2 où Q1 est le barycentre de PØ et P1 affectés des coefficients 1 et 3 et Q2 le barycentre de P2 et P1 affectés des coefficients 1 et 3. Les coordonnées de Q1 sont donc ([XØ + 3 * X1]/4, [YØ + 3 * Y1]/4) ; les coordonnées de Q2 sont donc ([X2 + 3 * X1]/4, [Y2 + 3 * Y1]/4) (voir lignes 310, 315).



Ligne 50

Initialisation. Voir le chapitre 0.

Ligne 100

Fixation de l'ordre du dragon. Attention pour N = 10 il y a 2^{10} = 1 024 segments ; pour N = 14 il y a 2^{14} = 16 384 segments, et cela prend du temps !

Ligne 110

Déclaration du tableau qui va contenir le codage de la règle de pliage.

Ligne 120

Définition de la règle de pliage. On place des Ø pour « au-dessus » et des 1 pour « au-dessous ». Pour le premier dragon qui correspond à la règle « au-dessus, au-dessus etc. », on a donc A(Ø) = Ø, A(1) = Ø etc.

Ligne 130

On fixe les coordonnées du premier point ; on fixe l'angle du premier segment dessiné avec l'horizontale ; on fixe la longueur des segments du dragon (sans les arrondis).

Lignes 140, 150

Initialisation des coordonnées des points PØ, P1, P2 et positionnement de la pointe.

Ligne 190

Calcul du nombre de segments du dragon.

Ligne 200

Début de la boucle principale. A chaque parcours de la boucle, on traite un segment du dragon, c'est-à-dire (à cause de l'arrondi) on dessine deux petits segments.

Lignes 215, 220, 230

Calculs permettant de savoir si on doit tourner à droite ou à gauche. Pour des explications plus précises consulter les références déjà indiquées.

Attention certains BASIC ne possèdent pas la fonction MOD. Si c'est le cas de votre BASIC sachez que :

A MOD B est exactement équivalent à
A - INT (A/B) * B

Le programme DRAGONS a été traduit pour l'Apple II (voir annexe) qui ne dispose pas de la fonction MOD. Regardez comment ont été traduites les lignes 220 et 230.

Lignes 240, 300

Calcul des coordonnées de P2 et mise à jour des coordonnées de PØ et P1.

Lignes 310, 320

Tracé des deux segments composant l'arrondi, en utilisant les formules de barycentre expliquées au-dessus.

Lignes 330, 1000

Fin de la boucle principale. Fin du programme.

REMARQUES

1. Si vous essayez d'autres règles que celles données dans les exemples, vous aurez sans doute des problèmes de cadrage et de centrage. Cela est inévitable à moins

   a) de mettre tous les calculs en mémoire avant de débuter le dessin, mais alors vous atteindrez vite les limites de votre micro et vous ne pourrez sans doute pas faire N = 14,

   b) de faire exécuter les calculs une première fois de façon à déterminer la bonne fenêtre, puis une seconde fois pour faire exécuter le tracé, mais vous doublerez le temps d'exécution qui est déjà assez long.

   Il s'agit cependant d'intéressants exercices de programmation.

2. Lorsque nous définissons la règle de pliage, nous supposons que le tableau que nous remplissons est préinitialisé avec des zéros. Cette préinitialisation des tableaux (et des variables) est implicite dans la plupart des BASIC mais il y a quelques exceptions.

Programme de base « DRAGONS »

```
10 'DRAGONS
50 LPRINT CHR$(18):LPRINT"M0,-800":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=6
110 DIM A(N)
120 FOR I=0 TO N:A(I)=0:NEXT I
130 X0=NP/3:Y0=NP/2:A0=-PI/4*(N-2):L0=NP
/SQR(2)^N
140 X1=X0:Y1=Y0:X2=X0:Y2=Y0
150 LPRINT"M";INT(X0);",";INT(Y0)
190 NN=2^N-1
200 FOR I=0 TO NN
210 IF I=0 THEN 300
215 II=I:J=0
220 IF (II MOD 2)=0 THEN II=II/2:J=J+1:G
OTO 220
230 AA=(A(N-J)*2-1)*( ((II-1)/2 MOD 2)*2
-1)*PI/2
240 A0=A0+AA
300 X0=X1:Y0=Y1:X1=X2:Y1=Y2:X2=X2+L0*COS
(A0):Y2=Y2+L0*SIN(A0)
```

```
310 LPRINT"D";INT((X0+3*X1)/4);",";INT((
Y0+3*Y1)/4)
315 LPRINT"D";INT((X2+3*X1)/4);",";INT((
Y2+3*Y1)/4)
330 NEXT I
1000 END
```

Modifications pour le dessin 50

```
10 'DRAGONS
100 N=6
```

Modifications pour le dessin 51

```
10 'DRAGONS
100 N=10
```

Modifications pour le dessin 52

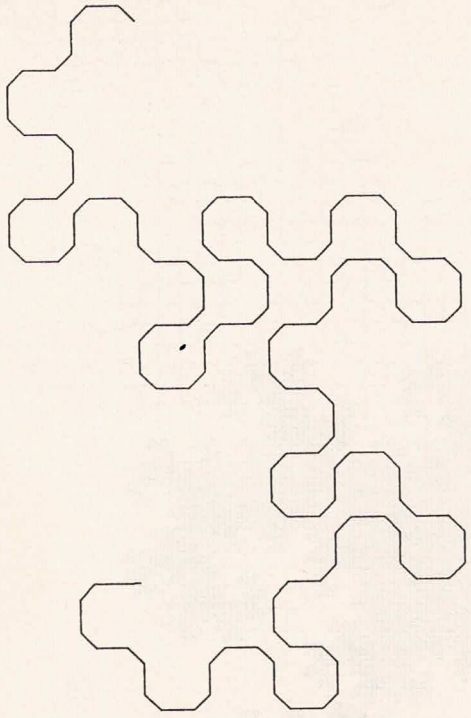
```
10 'DRAGONS
100 N=14
```

Modifications pour le dessin 53

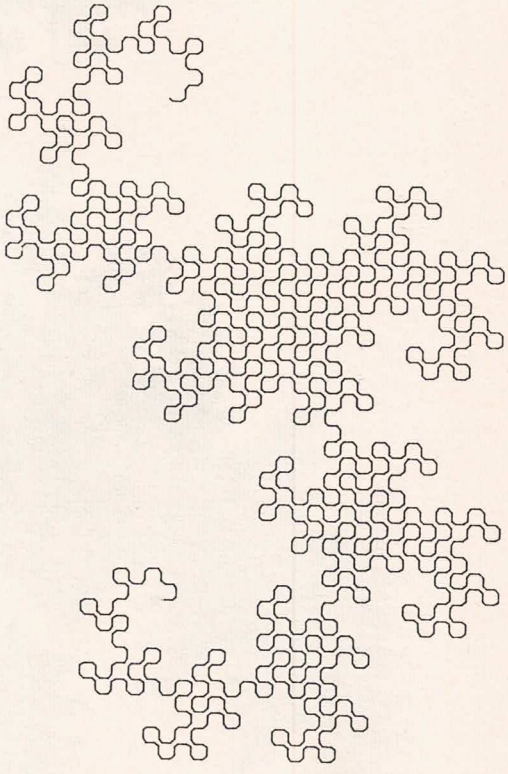
```
10 'DRAGONS
100 N=10
120 FOR I=0 TO N STEP 3:A(I)=1:NEXT I
130 X0=NP*.25:Y0=0:A0=-PI/2:L0=NP/SQR(2)
^N*.9
```

Modifications pour le dessin 54

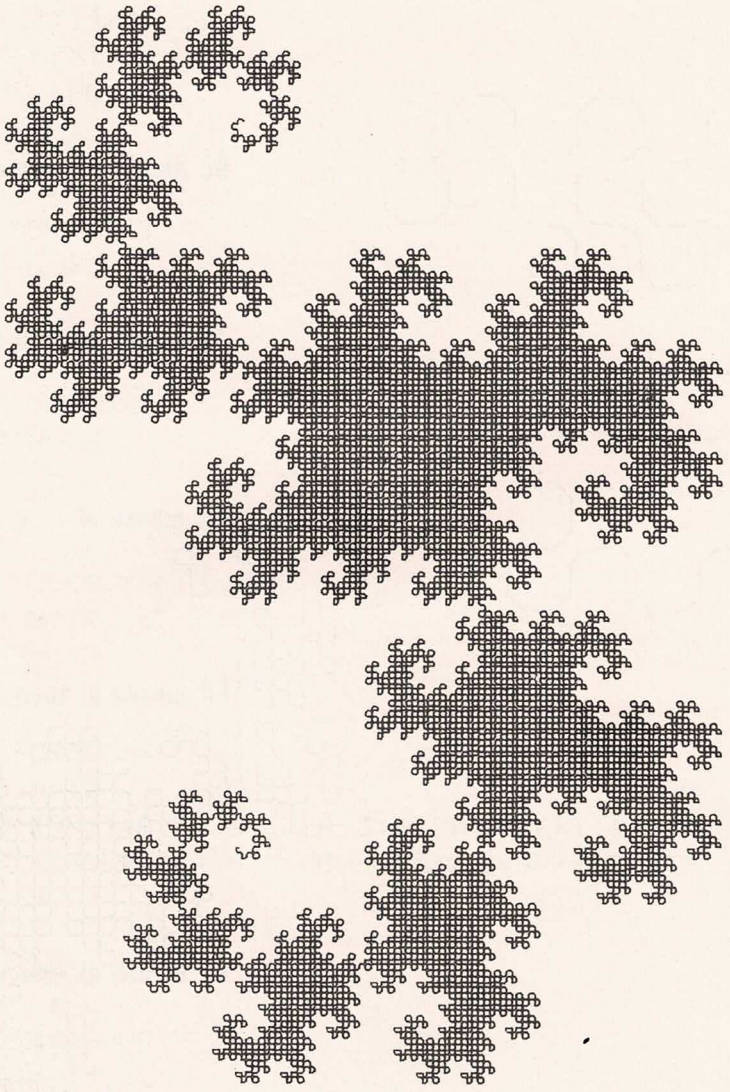
```
10 'DRAGONS
100 N=14
120 FOR I=0 TO N STEP 3:A(I)=1:NEXT I
130 X0=NP*.25:Y0=0:A0=-PI:L0=NP/SQR(2)^N
*.9
```



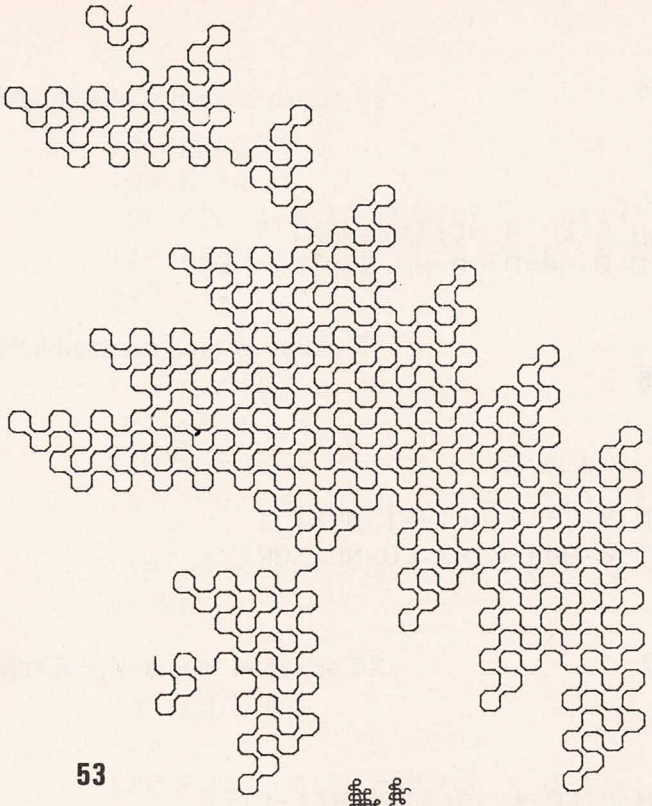
50



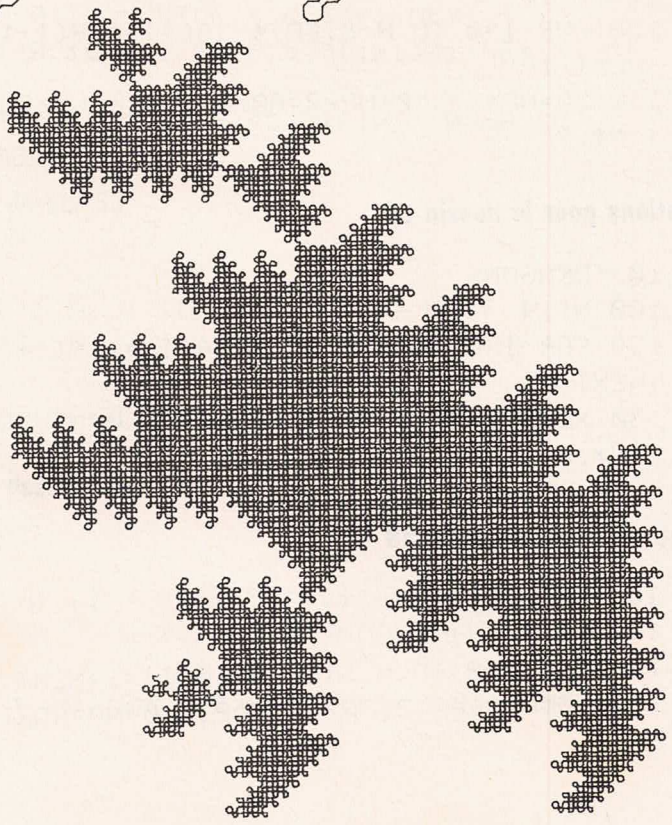
51



52



53



54

Modifications pour le dessin 55

```
10 'DRAGONS
100 N=10
120 FOR I=0 TO N STEP 4:A(I)=1:NEXT I
130 X0=NP*.62:Y0=0:A0=0:L0=NP/SQR(2)^N*.
9
```

Modifications pour le dessin 56

```
10 'DRAGONS
100 N=14
120 FOR I=0 TO N STEP 4:A(I)=1:NEXT I
130 X0=NP*.62:Y0=0:A0=-PI/2:L0=NP/SQR(2)
^N*.9
```

Modifications pour le dessin 57

```
10 'DRAGONS
100 N=10
120 FOR I=0 TO N STEP 4 :A(I)=1:A(I+1)=1
:NEXT I
130 X0=NP*.7:Y0=NP/2:A0=PI/2:L0=NP/SQR(2
)^N*.8
```

Modifications pour le dessin 58

```
10 'DRAGONS
100 N=14
120 FOR I=0 TO N STEP 4 :A(I)=1:A(I+1)=1
:NEXT I
130 X0=NP*.7:Y0=NP/2:A0=PI/2:L0=NP/SQR(2
)^N*.8
```

Modifications pour le dessin 59

```
10 'DRAGONS
100 N=10
120 FOR I=0 TO N STEP 5 :A(I)=1:NEXT I
130 X0=NP*.4:Y0=NP/2:A0=PI:L0=NP/SQR(2)^
N*.9
```

Modifications pour le dessin 60

```
10 'DRAGONS
100 N=14
120 FOR I=0 TO N STEP 5 :A(I)=1:NEXT I
130 X0=NP*.4:Y0=NP/2:A0=0:L0=NP/SQR(2)^N
*.9
```

Modifications pour le dessin 61

```
10 'DRAGONS
100 N=10
120 FOR I=0 TO N STEP 5:A(I)=1:NEXT I
122 FOR I=0 TO N-1 STEP 5:A(I+1)=1:NEXT
I
130 X0=NP*.55:Y0=NP/2:A0=0:L0=NP/SQR(2)^
N*.96
```

Modifications pour le dessin 62

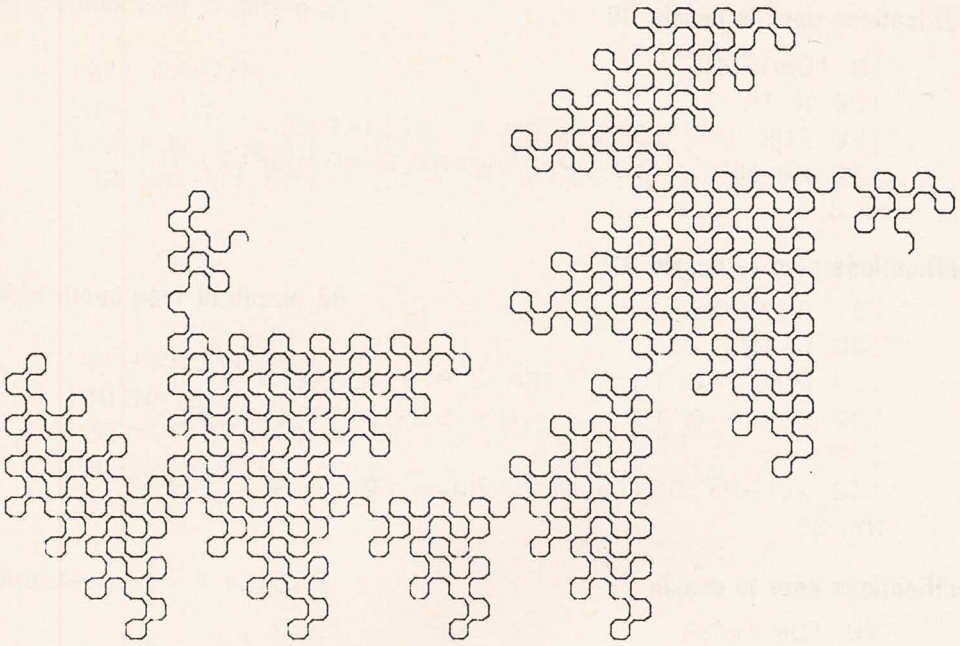
```
10 'DRAGONS
100 N=14
120 FOR I=0 TO N STEP 5:A(I)=1:NEXT I
122 FOR I=0 TO N-1 STEP 5:A(I+1)=1:NEXT
I
130 X0=NP*.55:Y0=NP/2:A0=3*PI/2:L0=NP/SQ
R(2)^N*.96
```

Modifications pour le dessin 63

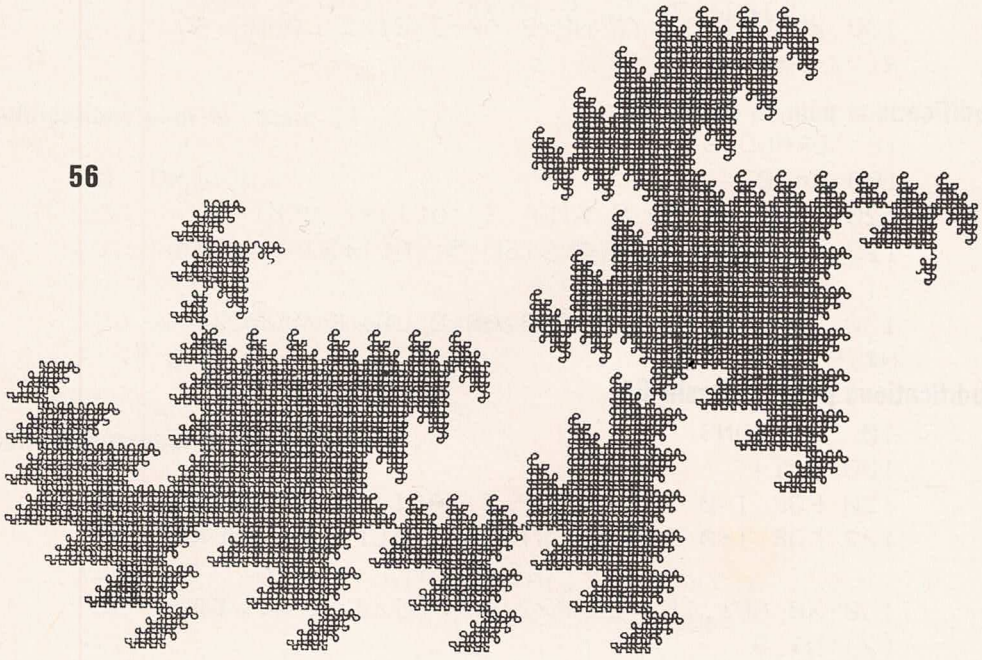
```
10 'DRAGONS
100 N=10
120 FOR I=0 TO N STEP 5 :A(I)=1:NEXT I
122 FOR I=0 TO N-2 STEP 5 :A(I+2)=1:NEXT
 I
130 X0=NP*.15:Y0=NP/2:A0=0:L0=NP/SQR(2)^
N*.9
```

Modifications pour le dessin 64

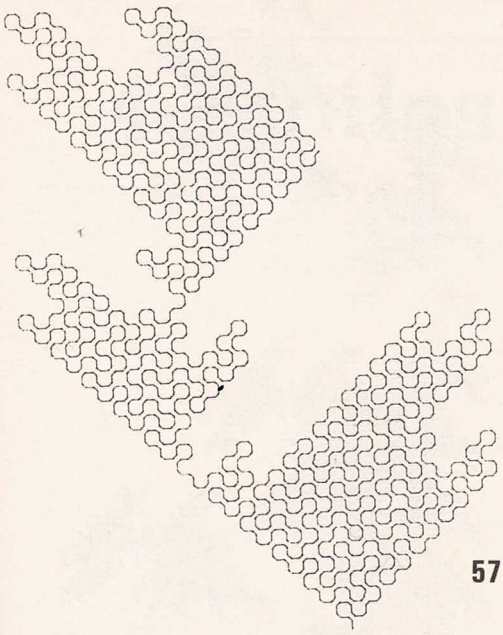
```
10 'DRAGONS
100 N=14
120 FOR I=0 TO N STEP 5 :A(I)=1:NEXT I
122 FOR I=0 TO N-2 STEP 5 :A(I+2)=1:NEXT
 I
130 X0=NP*.15:Y0=NP/2:A0=-PI/2:L0=NP/SQR
(2)^N*.9
```



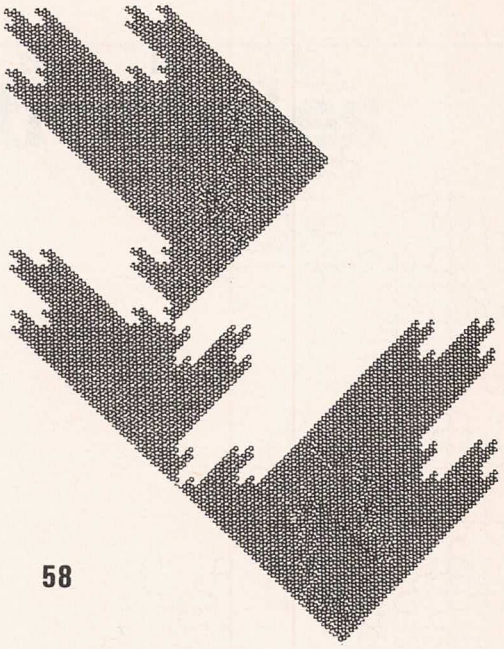
55



56

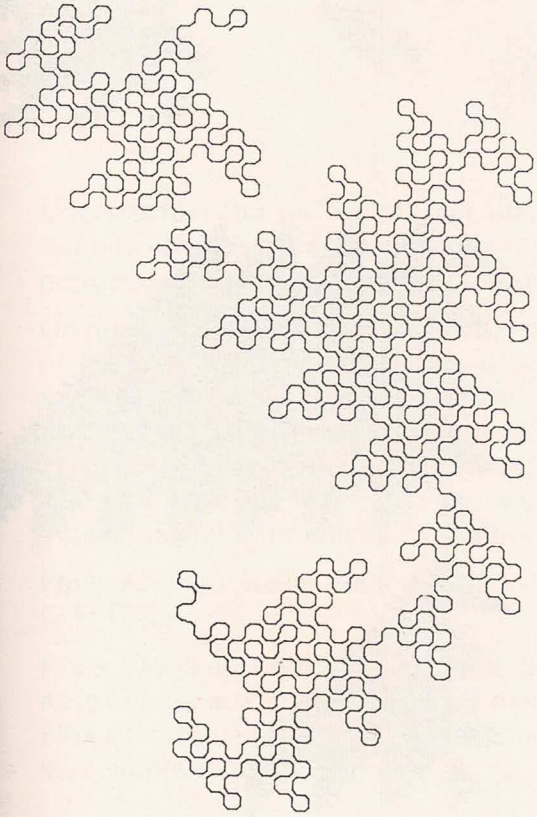


57

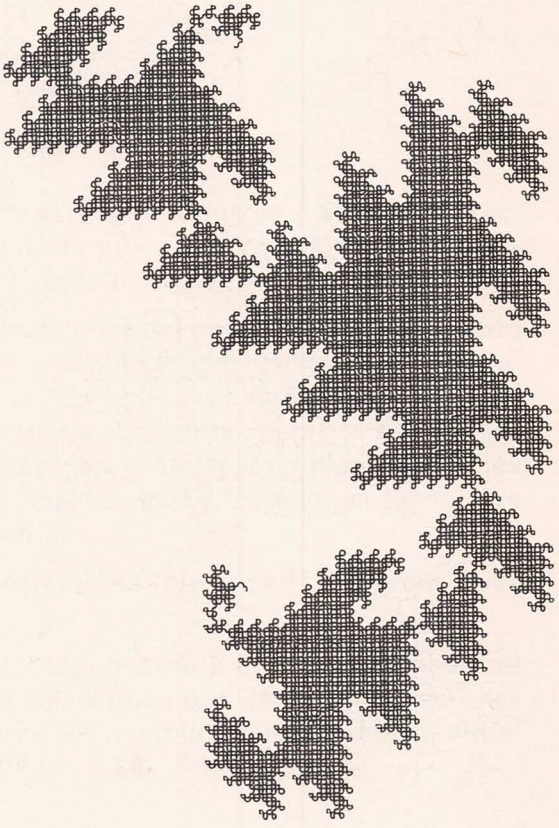


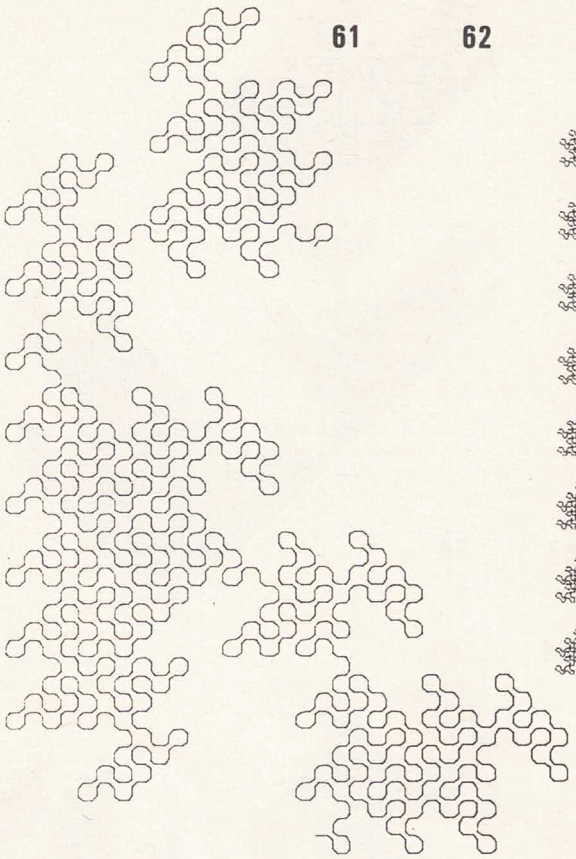
58

59

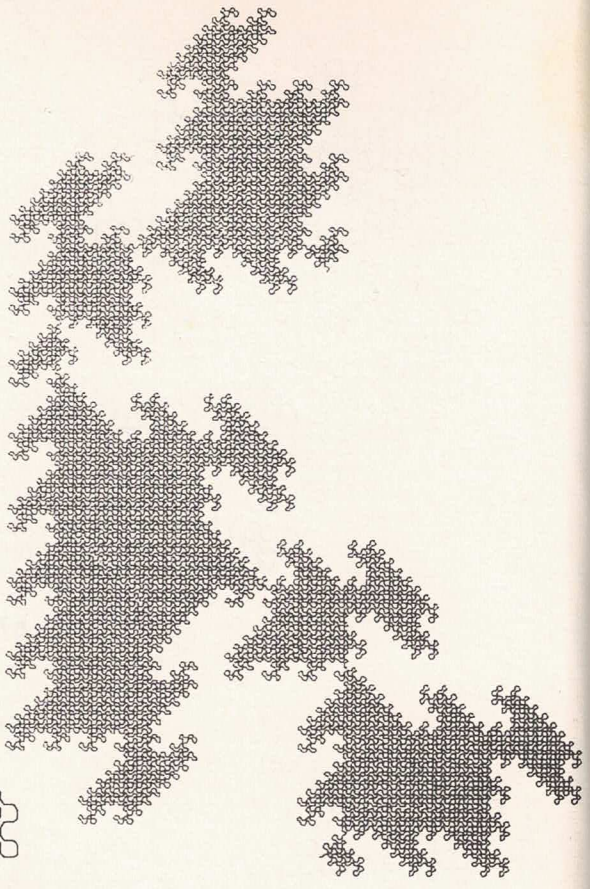


60

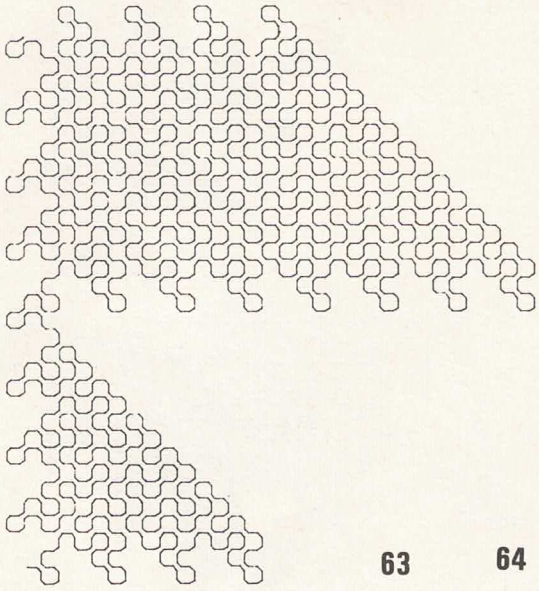




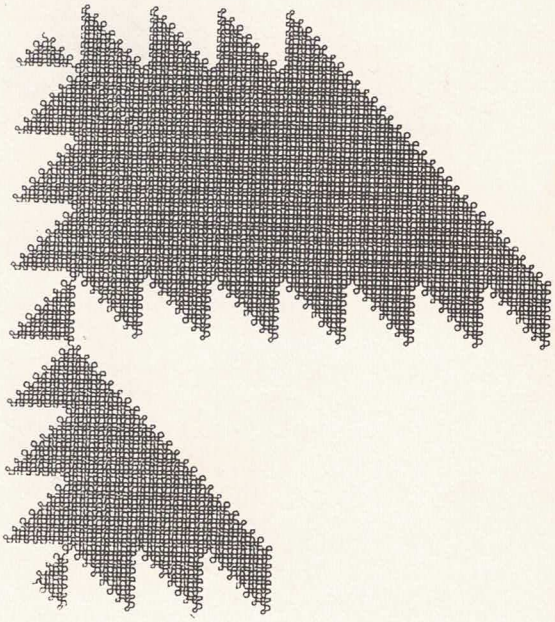
61



62



63



64

4 Etoiles fractales

Une fractale (ou un fractal) est un objet géométrique profondément fractionné, composé d'éléments infiniment complexes et ramifiés qui contiennent souvent dans leur structure tout l'objet lui-même à une échelle réduite.

On dit parfois qu'une fractale est quelque chose qui ne se simplifie pas quand on le regarde de près. Une raison d'ordre mathématique justifie aussi la dénomination de fractale : la dimension de ces objets mesurée d'une certaine façon n'est pas un entier (0 pour les points, 1 pour les courbes comme les droites, les cercles, les sinusoïdes, 2 pour les morceaux de plan comme les bandes, les disques, 3 pour les morceaux d'espace comme les pavés, les boules) mais intermédiaire : fractionnaire.

Pour plus de détails, vous pouvez consulter les références bibliographiques [24] [25].

Nous avons déjà rencontré des fractales au chapitre 2 et nous en rencontrerons encore aux chapitres 7 et 9 mais celles que nous décrivons ici sont les plus simples de toutes et l'algorithme qui les engendre possède des potentialités surprenantes.

Le programme ÉTOILES FRACTALES

L'idée de l'algorithme est la suivante :

on fait dessiner une suite de segments S(0), S(1),..., S(I),... tous orientés les uns par rapport aux autres d'un même angle AA, et dont la longueur dépend du numéro I du segment. On compte le nombre de zéros qui terminent I, lorsqu'on le décompose dans la base N−1. Si ce nombre est K−1, la longueur de S(I) est LL; si ce nombre est K−2, la longueur de S(I) est LL * RA ; si ce nombre est K−3, la longueur de S(I) est LL * RA * RA etc. Le nombre total de segments tracés est N * (N−1) ^ (K−1) (pour que le dessin se termine là où il a commencé).

Les paramètres qui interviennent sont donc N, K, LL, RA, AA. Ils sont fixés à la ligne 100 qui est la seule ligne à modifier pour passer d'un dessin à un autre.

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres selon les indications données.

Ligne 110

Calcul des coordonnées du point de départ XØ et YØ, et de l'angle du premier segment avec l'horizontale : AØ.

La ligne 110 que nous proposons permet le centrage de tous les exemples dans la fenêtre [O, NP] x [−NP, + NP].

Si vous essayez d'autres exemples ou si vous voulez obtenir un meilleur centrage, il faut jouer sur les valeurs de XØ, YØ et de LL comme nous l'avons fait pour la traduction du programme ÉTOILES FRACTALES pour l'Apple II (voir annexe). Une stratégie efficace consiste à fixer d'abord LL = .2 * NP, XØ = NP/2, YØ = NP/2 puis en fonction du résultat obtenu à déterminer de meilleures valeurs pour LL, XØ, YØ.

Ligne 120

Positionnement de la plume au point de départ.

Ligne 190

Calcul du nombre de segments à tracer.

Ligne 200

Début de la boucle principale. A chaque parcours de cette boucle, un segment est dessiné.

Lignes 210, 300

Calcul du nombre de zéros qui terminent I quand on décompose I en base N−1. Ce nombre est H.

On utilise la fonction MOD. Si vous ne disposez pas de cette fonction, voyez les explications données pour le programme DRAGONS (Chap. 3).

Lignes 310, 320

Calcul de la longueur LØ et de l'angle AØ du segment S(I) avec l'horizontale.

Lignes 330, 340

Calcul de la position de l'extrémité terminale du segment S(I).

Ligne 350

Tracé du segment S(I).

Lignes 400, 1 000

Fin de la boucle principale. Fin du programme.

Le dessin 65 est une étoile fractale à l'ordre 5 (l'ordre est donné par K). C'est-à-dire qu'il y a

5 fois
- une étoile avec à l'extrémité de chacune de ses branches
- une autre étoile qui elle-même possède à l'extrémité de chacune de ses branches
- une autre étoile qui elle-même...
- . . .
- . . .

Le dessin 66 montre une autre étoile fractale, cette fois-ci à l'ordre 4.

Pour le dessin 67, nous avons choisi RA = .5 et des étoiles à 12 branches ; ce qui donne un étonnant napperon.

Les dessins 68 à 73 fournissent des exemples élémentaires et moins élémentaires de figures géométriques engendrées par modification de la ligne 100.

Les dessins 74 - 77 sont des exemples de fractales dessinées à des ordres suffisamment grands pour qu'on arrive bien à imaginer le véritable objet fractal (qui ne s'obtient qu'à la limite lorsque l'ordre tend vers l'infini).

Profitez de ce programme simple pour vous créer une fractale personnelle !

Programme de base « ÉTOILES FRACTALES »

```
10 ' ETOILES FRACTALES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=5:K=5:RA=.35:LL=NP:AA=4*PI/5
110 X0=(NP-LL)/2:Y0=NP/4:A0=-AA
120 LPRINT"M";INT(X0);",";INT(Y0)
190 NN=N*(N-1)^(K-1)-1
200 FOR I= 0 TO NN
210 I1=I:H=0
300 IF (I1 MOD (N-1))=0 AND H<K-1 THEN I
1=I1/(N-1):H=H+1:GOTO 300
310 L0=LL*RA^(K-1-H)
320 A0=A0+AA
330 X0=X0+L0*COS(A0)
340 Y0=Y0+L0*SIN(A0)
350 LPRINT"D";INT(X0);",";INT(Y0)
400 NEXT I
1000 END
```

Modifications pour le dessin 65

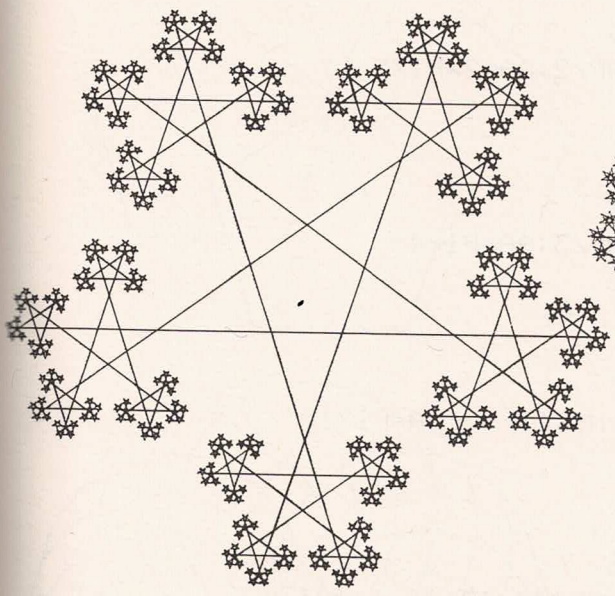
```
10 ' ETOILES FRACTALES
```

Modifications pour le dessin 66

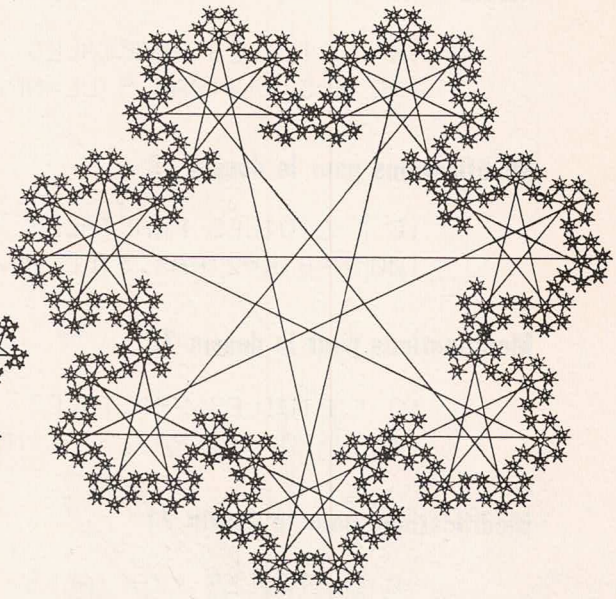
```
10 ' ETOILES FRACTALES
100 N=7:K=4:RA=.32:LL=NP:AA=6*PI/7
```

Modifications pour le dessin 67

```
10 ' ETOILES FRACTALES
100 N=12:K=3:RA=.5:LL=NP*11/48:AA=2*PI/1
2
```

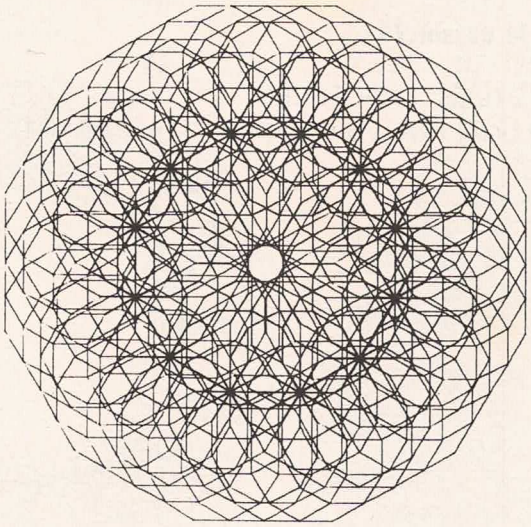


65



66

67



Modifications pour le dessin 68

```
10 ' ETOILES FRACTALES
100 N=5:K=2:RA=.5:LL=NP/2:AA=2*PI/5
```

Modifications pour le dessin 69

```
10 ' ETOILES FRACTALES
100 N=8:K=2:RA=.5:LL=NP/3:AA=PI/4
```

Modifications pour le dessin 70

```
10 ' ETOILES FRACTALES
100 N=20:K=2:RA=.5:LL=NP*13/96:AA=PI/10
```

Modifications pour le dessin 71

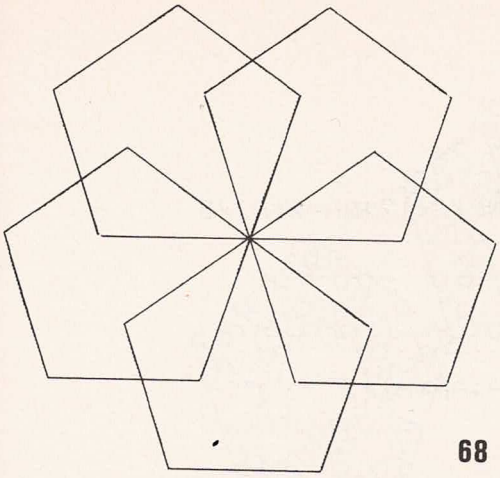
```
10 ' ETOILES FRACTALES
100 N=15:K=2:RA=.9:LL=NP*43/48:AA=14*PI/
15
```

Modifications pour le dessin 72

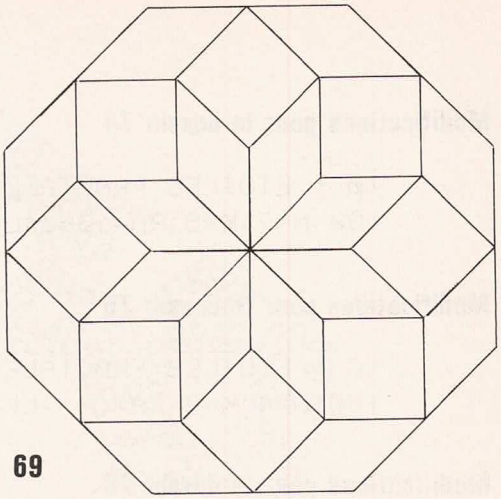
```
10 ' ETOILES FRACTALES
100 N=16:K=3:RA=.27:LL=NP/6:AA=PI/8
```

Modifications pour le dessin 73

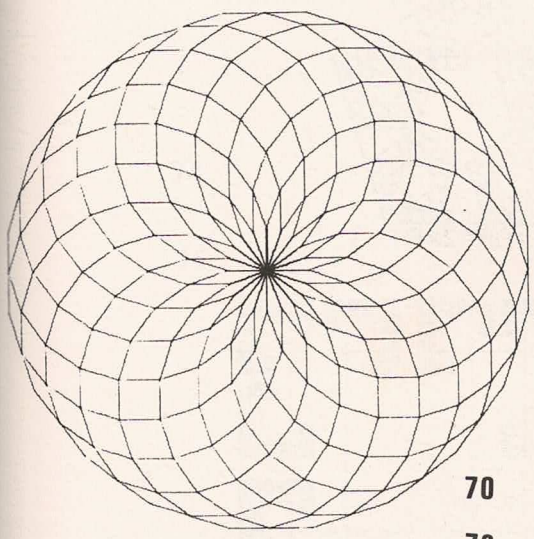
```
10 ' ETOILES FRACTALES
100 N=8:K=4:RA=.5:LL=NP*17/48:AA=PI/4
```



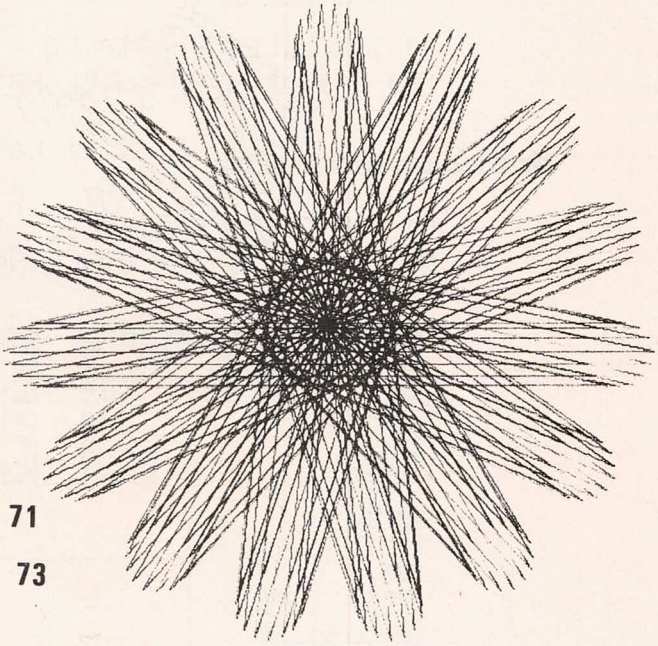
68



69

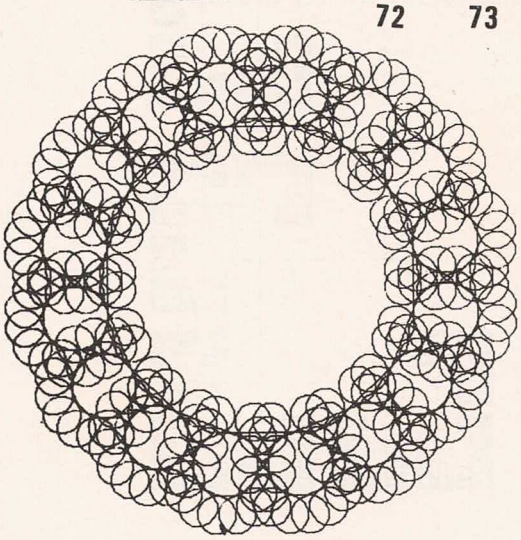


70

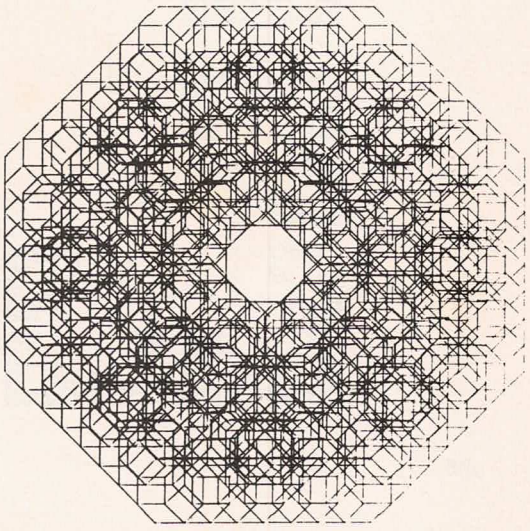


71

72



73



Modifications pour le dessin 74

```
10 ' ETOILES FRACTALES
100 N=7:K=5:RA=.383:LL=NP*7/12:AA=2*PI/5
```

Modifications pour le dessin 75

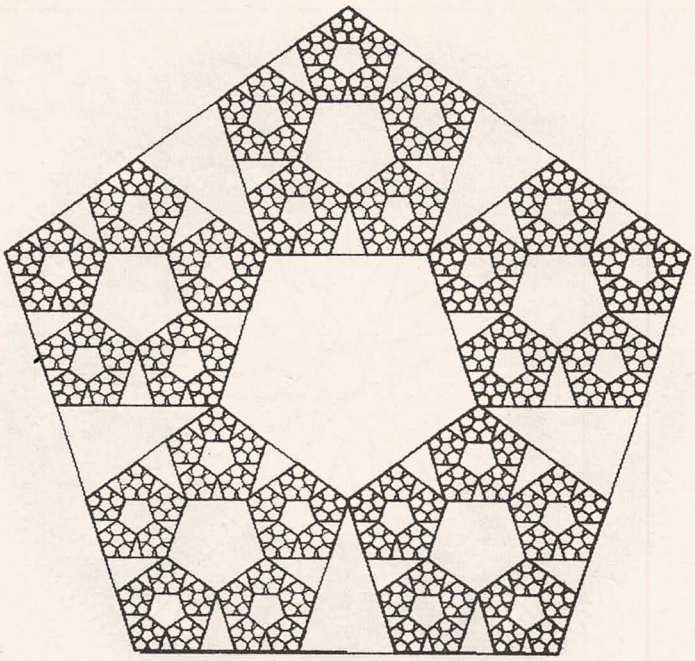
```
10 ' ETOILES FRACTALES
100 N=4:K=8:RA=.47:LL=NP:AA=PI/2
```

Modifications pour le dessin 76

```
10 ' ETOILES FRACTALES
100 N=15:K=3:RA=.3:LL=NP:AA=14*PI/15
```

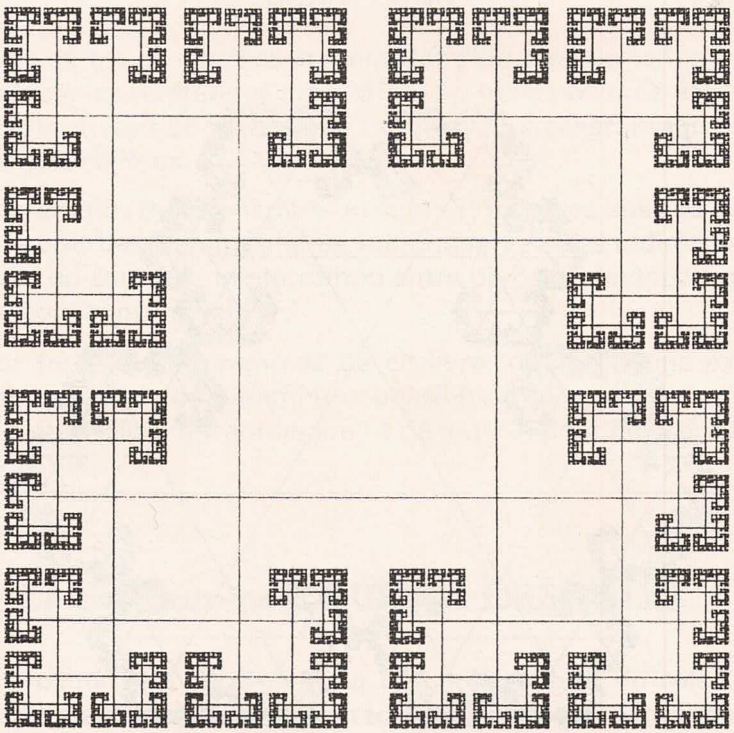
Modifications pour le dessin 77

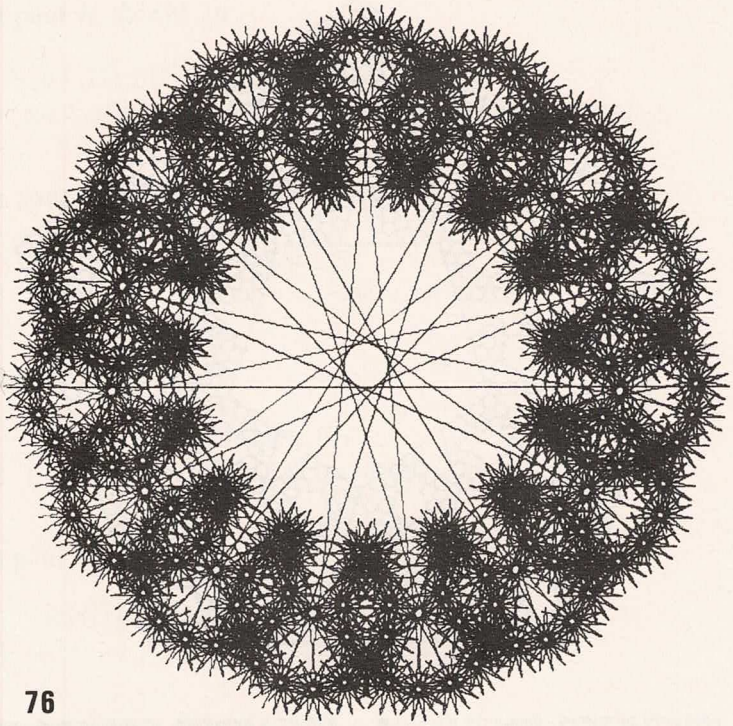
```
10 ' ETOILES FRACTALES
100 N=3:K=11:RA=.62:LL=NP:AA=2*PI/3
```



74

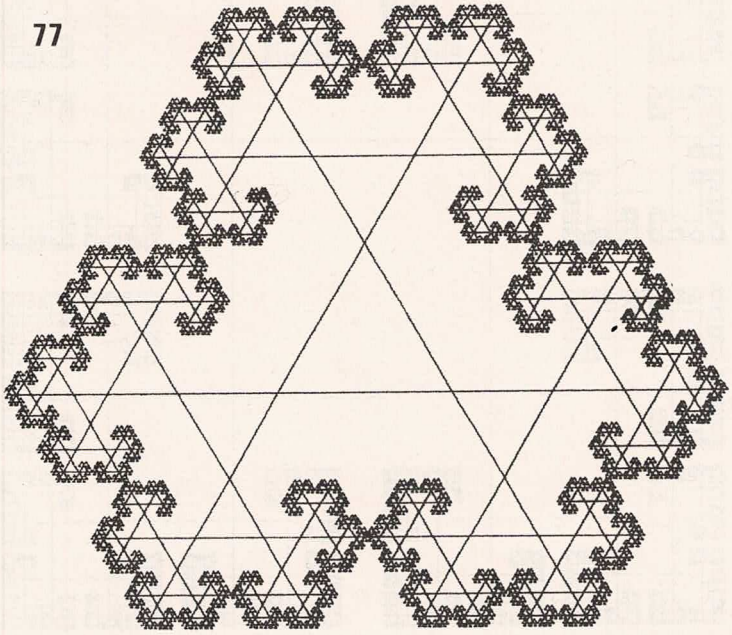
75





76

77



5 Courbes

Il existe toutes sortes de courbes et même les plus simples sont parfois belles (voir par exemple les références [1] [18] [23]). Mais nos buts étant délibérément décoratifs et esthétiques, nous ne nous intéresserons qu'à certaines familles très particulières.

Les trois programmes de ce chapitre, exécutent tous des dessins de ce qu'on appelle des « courbes paramétriques continues », c'est-à-dire des courbes obtenues avec un seul trait ininterrompu entre un point de départ et un point d'arrivée (parfois confondus).

Comme pour tous les programmes de ce livre, nous n'avons exploré que quelques-uns des choix de paramètres possibles ; il y a sans doute dans les variantes des merveilles extraordinaires à découvrir, pour qui sait être patient et astucieux.

Le programme COURBES ORBITALES

Imaginez une planète se déplaçant sur un cercle autour du soleil. Imaginez autour de cette planète un satellite qui tourne tout en se rapprochant de plus

en plus de la planète, en faisant par exemple 100 tours pendant que la planète en fait un autour du soleil, et de telle façon qu'après deux tours de la planète autour du soleil, le satellite et la planète se rejoignent. En dessinant la trajectoire du satellite, vous obtenez le dessin 78.

Bien que cela ne soit pas physiquement possible, on peut imaginer que la planète décrit une courbe plus compliquée et que le satellite s'approche et s'éloigne de la planète selon une règle différente. Ce sont toutes ces « trajectoires orbitales » qu'on peut visualiser avec ce programme.

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres :

N : nombre de segments de la courbe, car bien sûr, on approche la trajectoire par une suite de segments qui, s'ils sont assez petits, sont imperceptibles.

T1 : nombre de tours de la planète autour du soleil.

T2 : nombre de tours du satellite autour de la planète.

R1 : rayon de la courbe décrite par la planète.

K1, K2 : paramètres déterminant la forme de la courbe décrite par la planète ; avec K1 = 1, K2 = 1 on obtient un cercle.

Ligne 200

Début de la boucle principale. A chaque parcours de la boucle, un segment élémentaire est tracé. On peut considérer I comme une variable mesurant le temps.

Ligne 210

Calcul du rayon de l'orbite du satellite. Ce rayon est NP * .2 à l'instant I = 0, et est 0 à l'instant I = N.

Ligne 220

Calcul des angles qui déterminent la position de la planète (angle A1) et la position du satellite (angle A2).

Lignes 230, 240

Calcul des coordonnées du satellite à l'instant I.

Lignes 300, 310

Lorsque I = 0 positionnement de la plume.

Lorsque I > 0, tracé du segment entre la position du satellite à l'instant I — 1 et la position du satellite à l'instant I.

Lignes 400, 500

Fin de la boucle principale. Fin du programme.

Le dessin 78 a déjà été expliqué. Le dessin 79 représente encore la trajectoire d'un satellite en orbite circulaire autour d'une planète elle-même en orbite circulaire autour du soleil, mais cette fois-ci le satellite s'approche et s'éloigne périodiquement de la planète selon la formule de la ligne 210.

Pour le dessin 80, la trajectoire de la planète n'est plus une ellipse mais un huit (K1 = 1, K2 = 2).

Aux lignes 230 et 240 on a interverti les rôles de X % et de Y % et on a multiplié Y % par 1,5 de manière à obtenir une forme allongée.

D'autres trajectoires pour la planète (choix de K1 et K2) et d'autres lois d'éloignement-rapprochement (ligne 210) donnent les dessins 81, 82, 83, 84, 85, 86.

Programme de base « COURBES ORBITALES »

```
10 'COURBES ORBITALES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=2000:T1=2:T2=100:K1=1:K2=1:R1=NP*.
25
200 FOR I=0 TO N
210 R2=NP*.2*(1-I/N)
220 A1=2*PI*I/N*T1:A2=2*PI*I/N*T2
230 X%=INT(NP*.5+R1*COS(K1*A1)+R2*COS(A2
))
240 Y%=INT(NP*.5+R1*SIN(K2*A1)+R2*SIN(A2
))
300 IF I=0 THEN LPRINT"M";X%;",";Y%
310 IF I>0 THEN LPRINT"D";X%;",";Y%
400 NEXT I
500 END
```

Modifications pour le dessin 78

```
10 'COURBES ORBITALES
100 N=2000:T1=2:T2=100:K1=1:K2=1:R1=NP*.
25
```

Modifications pour le dessin 79

```
10 'COURBES ORBITALES
100 N=4000:T1=1:T2=200:K1=1:K2=1:R1=NP*.
2
210 R2=NP*.3*(.4*COS(2*I*PI/N*5)+.6)
```

Modifications pour le dessin 80

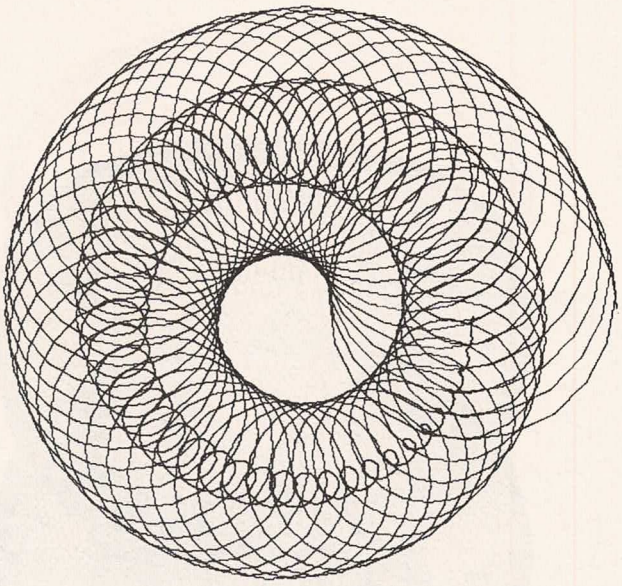
```
10 'COURBES ORBITALES
100 N=6000:T1=1:T2=300:K1=1:K2=2:R1=NP*.
3
210 R2=NP*.2*(.5+.5*COS(2*PI*I/10))
230 Y%=INT(1.5*(NP*.5+R1*COS(K1*A1)+R2*C
OS(A2)))
240 X%=INT(NP*.5+R1*SIN(K2*A1)+R2*SIN(A2
))
```

Modifications pour le dessin 81

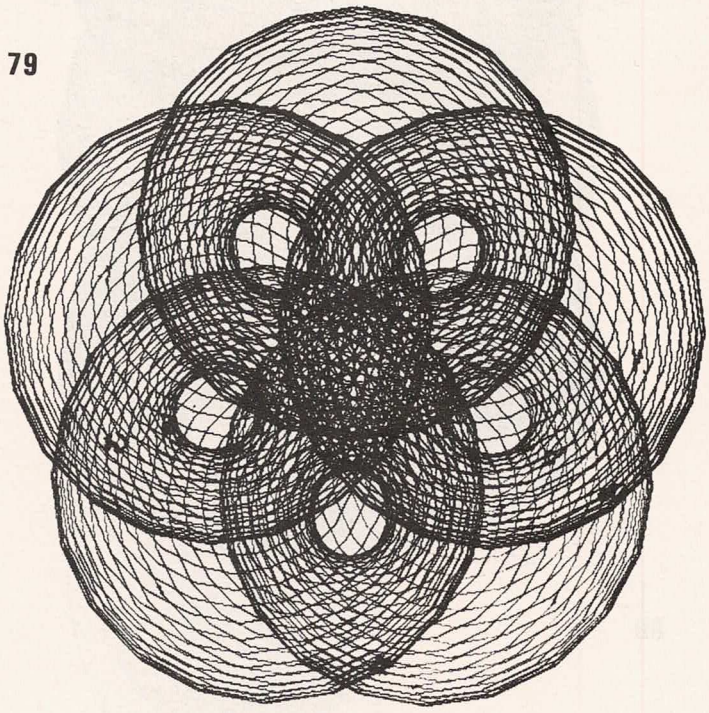
```
10 'COURBES ORBITALES
100 N=2000:T1=1:T2=100:K1=1:K2=1:R1=NP*.
27
210 R2=NP*.23*(.5*COS(2*I*PI/N*3)+.5)
240 Y%=INT(1.4*(NP*.5+R1*SIN(K2*A1)+R2*S
IN(A2)))
```

Modifications pour le dessin 82

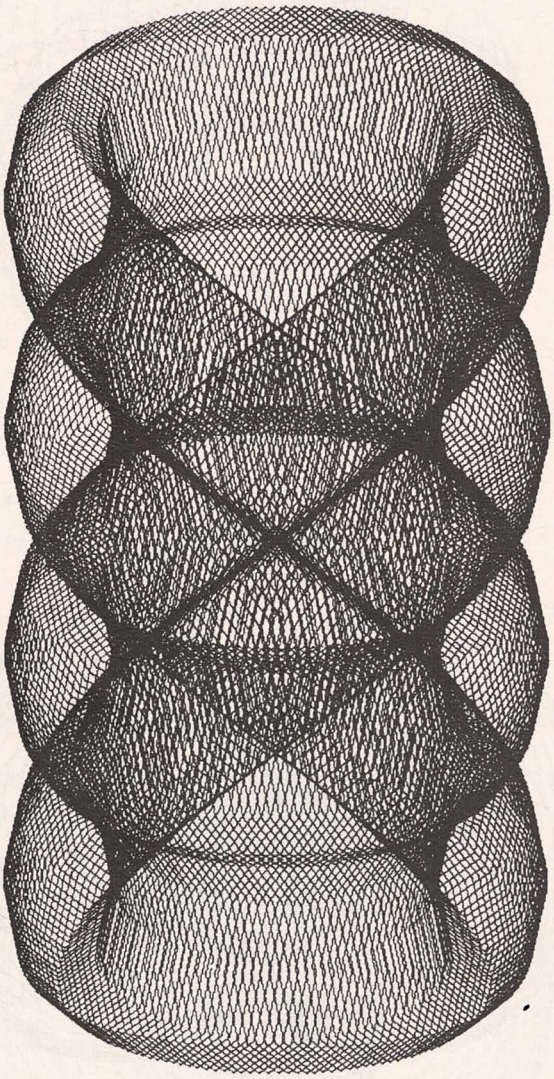
```
10 'COURBES ORBITALES
100 N=4000:T1=100:T2=1:K1=3:K2=2:R1=NP*.
1
210 R2=NP*.4*(1-I/N)
240 Y%=INT(1.6*(NP*.5+R1*SIN(K2*A1)+R2*S
IN(A2)))
```



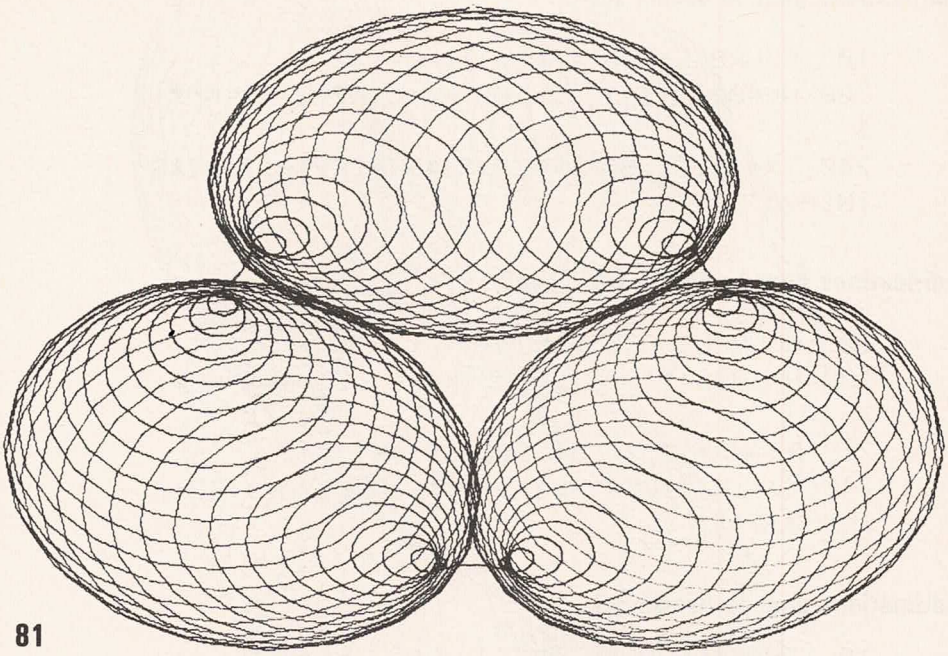
78



79

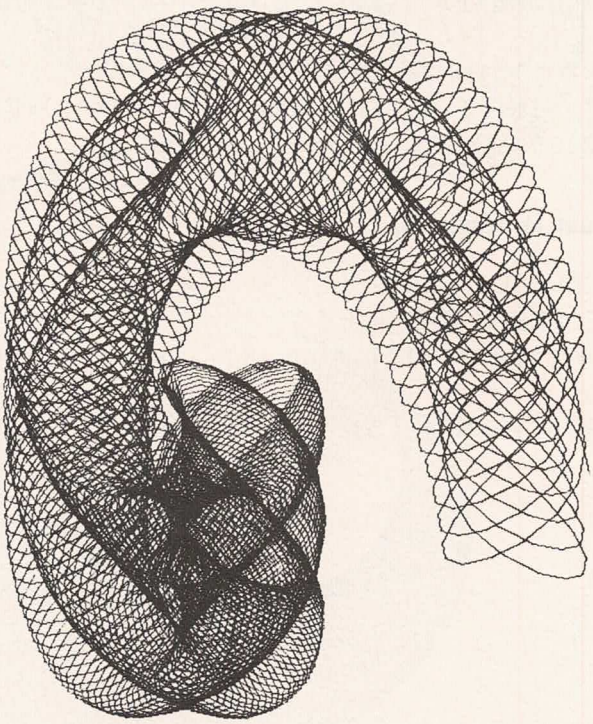


80



81

82



Modifications pour le dessin 83

```
10 'COURBES ORBITALES
100 N=4000:T1=1:T2=200:K1=3:K2=2:R1=NP*.
3
240 Y%=INT(1.5*(NP*.5+R1*SIN(K2*A1)+R2*S
IN(A2)))
```

Modifications pour le dessin 84

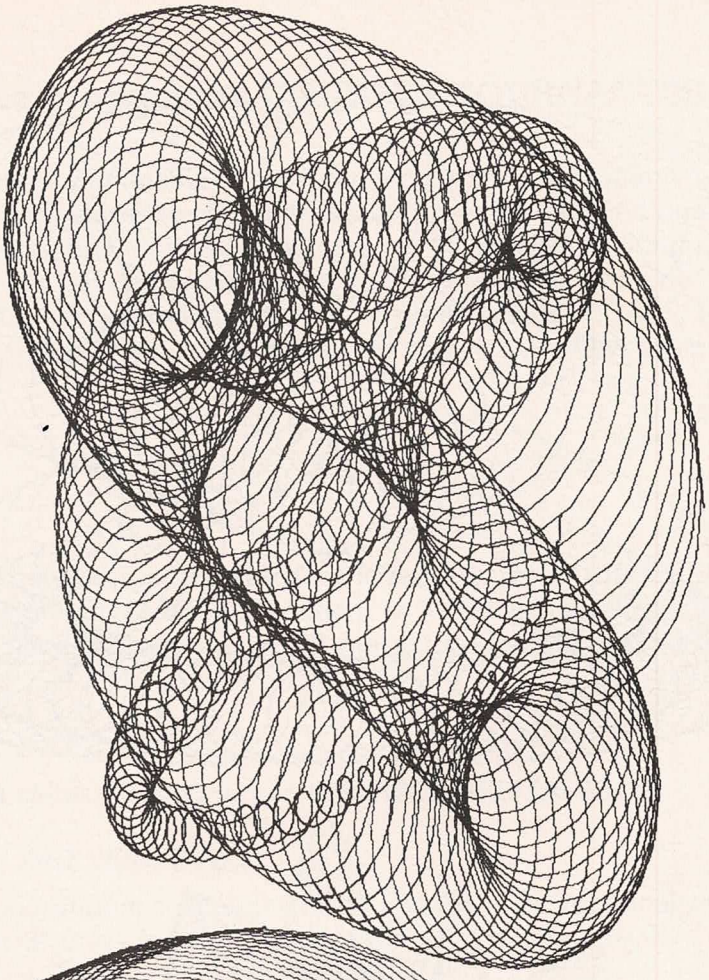
```
10 'COURBES ORBITALES
100 N=3000:T1=1:T2=150:K1=1:K2=1:R1=NP*.
25
210 R2=NP*.25*(.5+.5*COS(I*PI/N))
240 Y%=INT(1.3*(NP*.5+R1*SIN(K2*A1)+R2*S
IN(A2)))
```

Modifications pour le dessin 85

```
10 'COURBES ORBITALES
100 N=5000:T1=2:T2=250:K1=1:K2=2:R1=NP*.
35
210 R2=.15*NP*(.5+.5*COS(2*PI*I/N))
240 Y%=INT(1.4*(NP*.5+R1*SIN(K2*A1)+R2*S
IN(A2)))
```

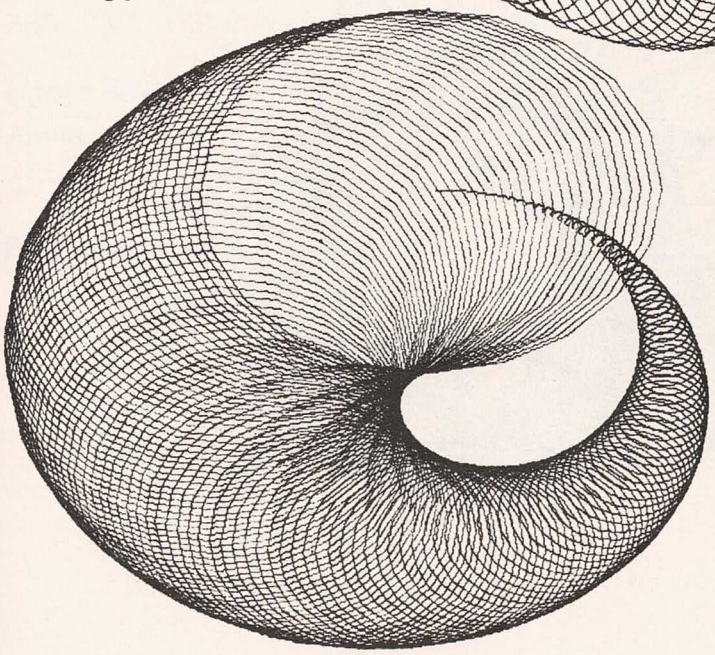
Modifications pour le dessin 86

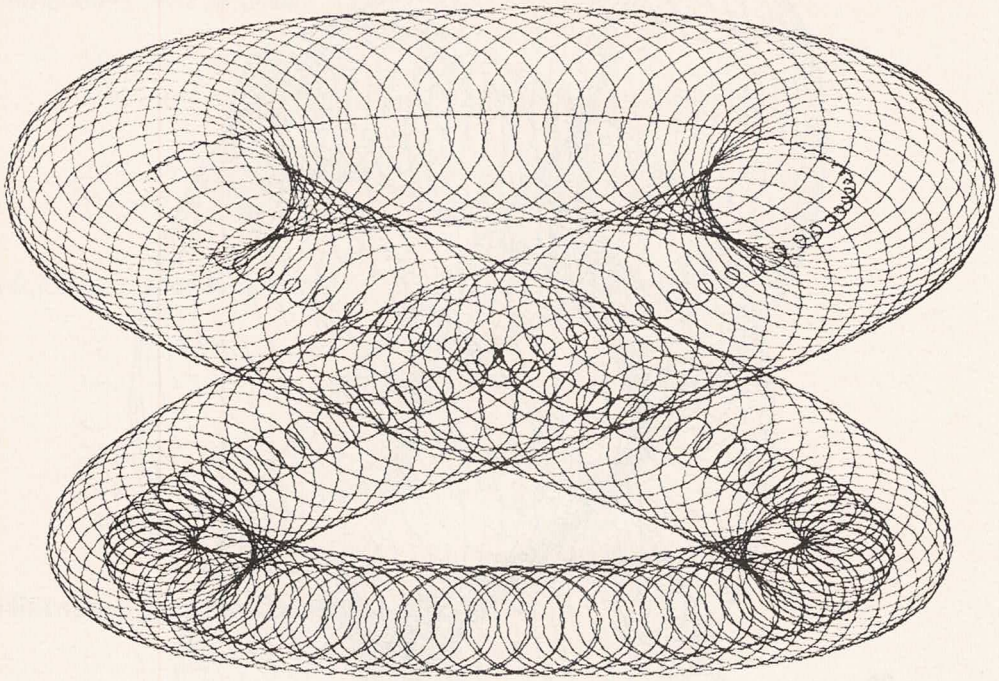
```
10 'COURBES ORBITALES
10 'COURBES ORBITALES
100 N=1400:T1=1:T2=600:K1=1:K2=1:R1=NP*.
25
210 R2=.25*NP*(.5+.5*COS(14*PI*I/N))
```



83

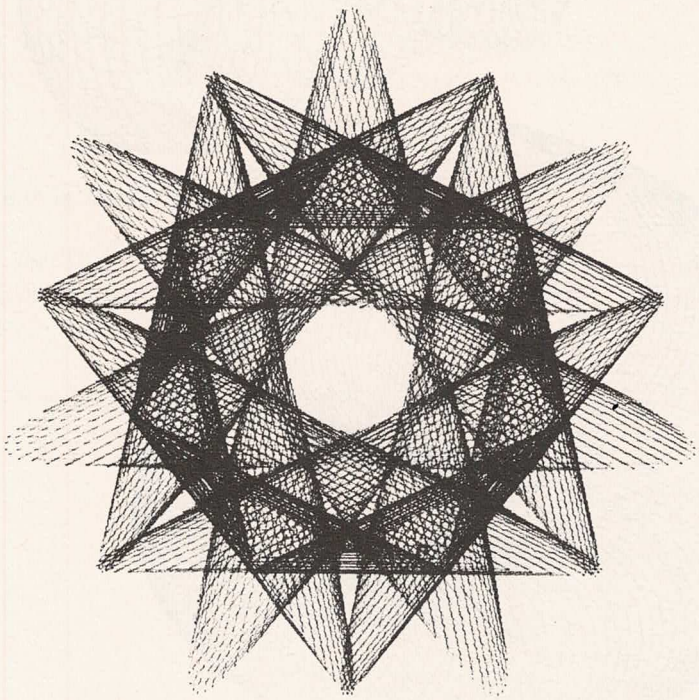
84





85

86



Le programme COURBES TOURNANTES

L'idée est assez semblable à celle du programme précédent : un satellite décrit une courbe dont les paramètres sont K1 et K2 autour d'une planète qui elle-même décrit une courbe (de paramètres H1 et H2) autour du soleil. Mais cette fois-ci, on fait pivoter la courbe du satellite en fonction de l'angle de la courbe de la planète, ceci de façon à ce qu'en fixant H1 = 1, H2 = 1, on obtienne un dessin bien centré autour du soleil comme le dessin 88.

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres, comme dans le programme COURBES ORBITALES. Il y a en plus H1 et H2 qui déterminent la forme de la courbe décrite par le satellite autour de la planète, et R2 qui est un rayon maximum pour cette courbe.

Lignes 200, 210, 220

Analogues à celles du programme COURBES ORBITALES.

Lignes 230, 240, 300, 310

Calcul des coordonnées du satellite à l'instant I. On reconnaîtra aux lignes 230, 240 et 300 les formules de rotation (voir le chapitre 2).

Lignes 400, 410, 500, 1 000

Analogues aux lignes 300, 310, 400, 500 du programme COURBES ORBITALES.

Programme de base « COURBES TOURNANTES »

```
10 'COURBES TOURNANTES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=2000:T1=1:T2=100:K1=1:K2=1:H1=1:H2
=1:R1=NP/6:R2=NP/4
200 FOR I=0 TO N
210 S=COS(4*PI*I/N)*.4+.6
220 AN=2*PI*I/N
230 C1=COS(H1*AN*T1):S1=SIN(H2*AN*T1)
```

```
240 C2=S*COS(K1*AN*T2):S2=S*SIN(K2*AN*T2
)
300 X=NP/2+R1*C1+R2*(C1*C2-S1*S2)
310 Y=NP/2+R1*S1+R2*(S1*C2+C1*S2)
400 IF I=0 THEN LPRINT"M";INT(X);",";INT
(Y)
410 IF I>0 THEN LPRINT"D";INT(X);",";INT
(Y)
500 NEXT I
1000 END
```

Modifications pour le dessin 87

```
10 'COURBES TOURNANTES
```

Modifications pour le dessin 88

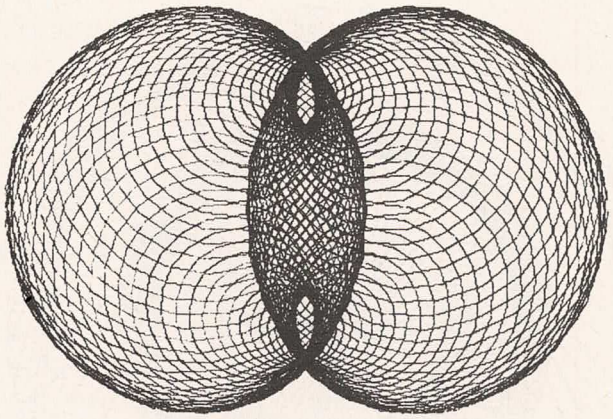
```
10 'COURBES TOURNANTES
100 N=3000:T1=1:T2=100:H1=1:H2=1:K1=1:K2
=2:R1=NP/3:R2=NP/7
210 S=1
```

Modifications pour le dessin 89

```
10 'COURBES TOURNANTES
100 N=3000:T1=.5:T2=30:H1=1:H2=1:K1=1:K2
=3:R1=NP/8:R2=NP*.27
210 S=1
```

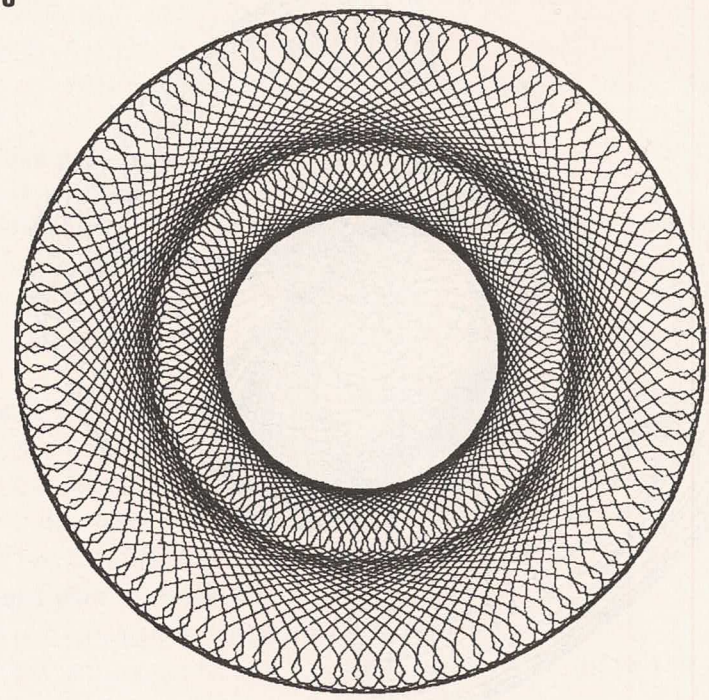
Modifications pour le dessin 90

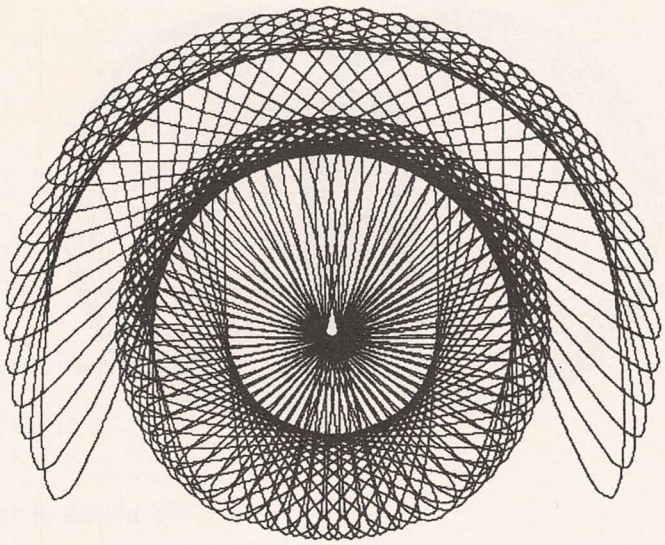
```
10 'COURBES TOURNANTES
100 N=2000:T1=.5:T2=50:H1=1:H2=2:K1=1:K2
=2:R1=NP/7:R2=NP/4
210 S=1
```



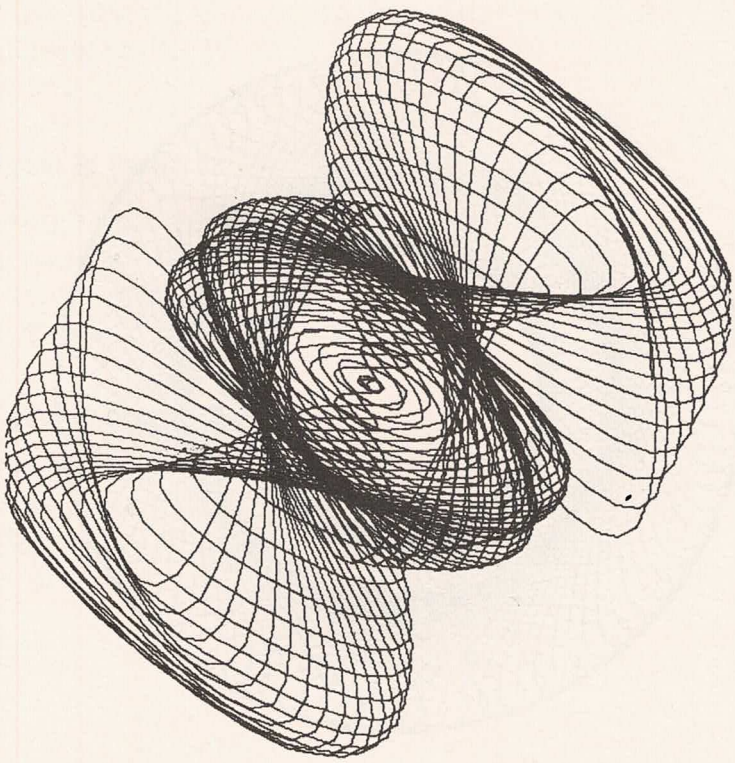
87

88





89



90

Modifications pour le dessin 91

```
10 'COURBES TOURNANTES
100 N=4000:T1=1:T2=800:H1=3:H2=5:K1=1:K2
=1:R1=NP*5/12:R2=NP/24
210 S=COS(30*PI*I/N)*.4+.6
310 Y=NP/2+R1*S1+R2*(S1*C2+C1*S2):Y=1.4*
Y
```

Modifications pour le dessin 92

```
10 'COURBES TOURNANTES
100 N=4000:T1=1:T2=100:H1=1:H2=1:K1=1:K2
=2:R1=NP/6:R2=NP/4
210 S=COS(3*PI*I/N)*.5+.5
310 Y=NP/2+R1*S1+R2*(S1*C2+C1*S2):Y=1.6*
Y
```

Modifications pour le dessin 93

```
10 'COURBES TOURNANTES
100 N=3000:T1=1:T2=100:H1=1:H2=2:K1=1:K2
=2:R1=NP/3:R2=NP/8
210 S=COS(8*PI*I/N)*.4+.6
310 Y=NP/2+R1*S1+R2*(S1*C2+C1*S2):Y=1.3*
Y
```

Modifications pour le dessin 94

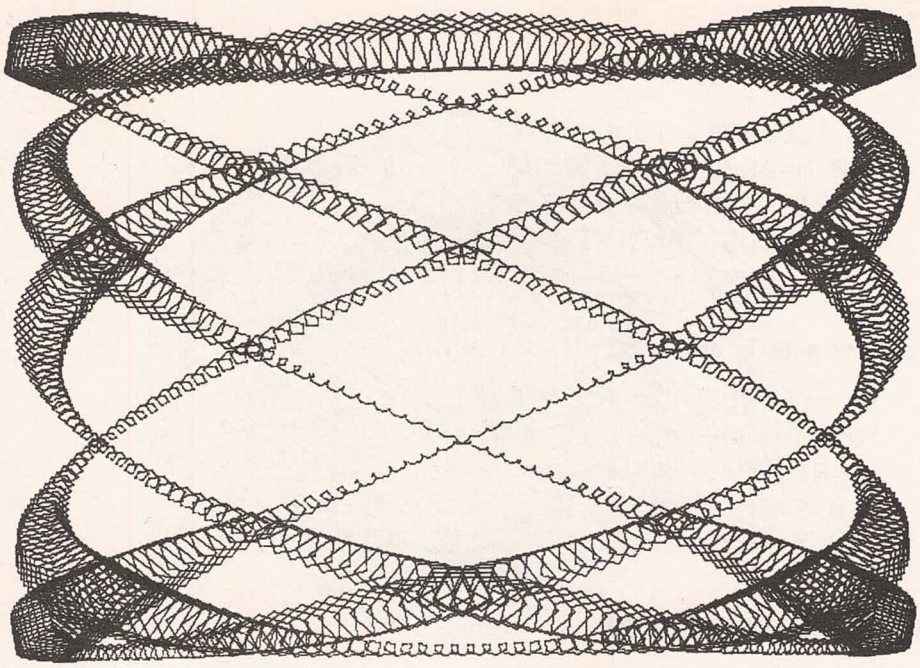
```
10 'COURBES TOURNANTES
100 N=5000:T1=1:T2=200:H1=1:H2=2:K1=1:K2
=2:R1=NP*3/8:R2=NP/12
210 S=COS(18*PI*I/N)*.4+.6
310 Y=NP/2+R1*S1+R2*(S1*C2+C1*S2):Y=1.5*
Y
```

Modifications pour le dessin 95

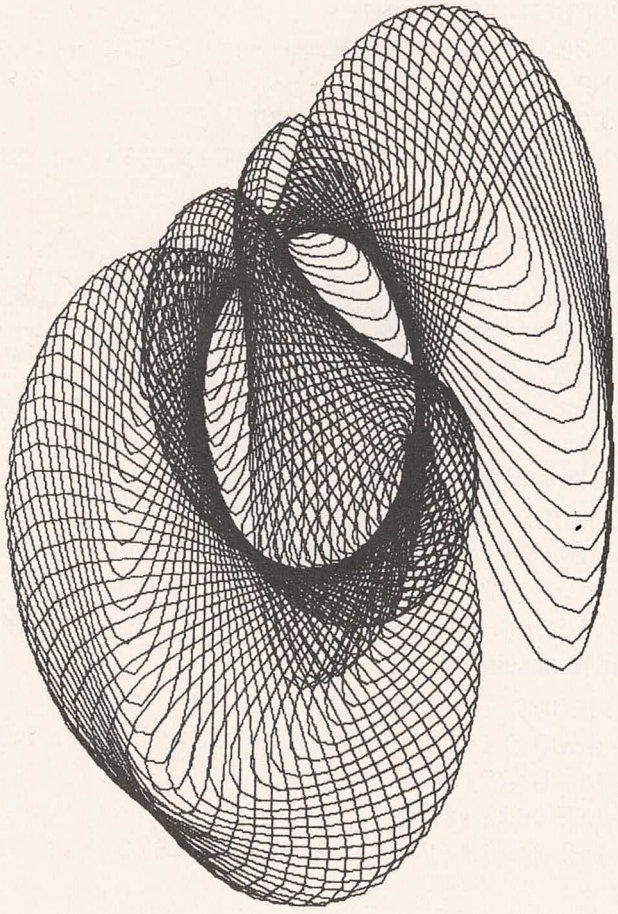
```
10 'COURBES TOURNANTES
100 N=4000:T1=1:T2=100:H1=1:H2=1:K1=1:K2
=2:R1=NP/6:R2=NP/4
210 S=COS(4*PI*I/N)*.4+.6
```

Modifications pour le dessin 96

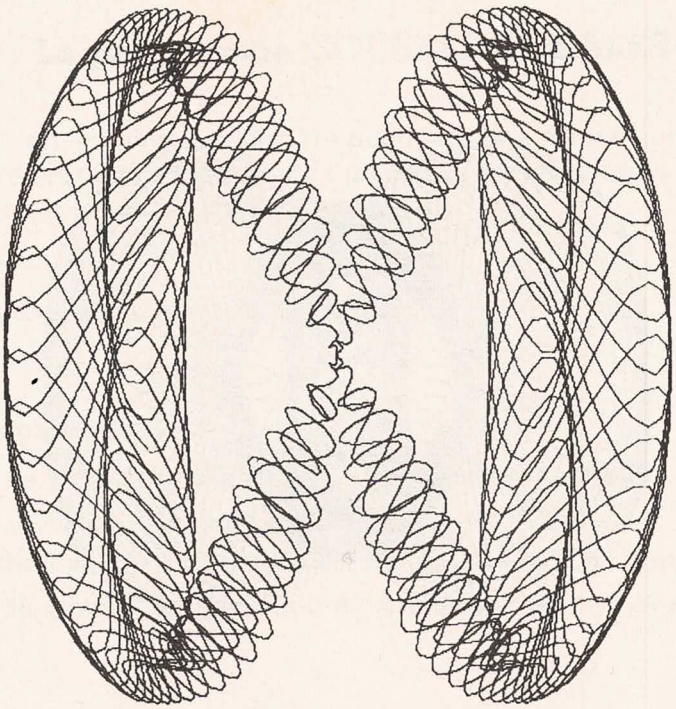
```
10 'COURBES TOURNANTES
100 N=4000:T1=1:T2=100:H1=1:H2=1:K1=1:K2
=2:R1=NP/6:R2=NP/4
210 S=COS(6*PI*I/N)*.4+.6
310 Y=NP/2+R1*S1+R2*(S1*C2+C1*S2):Y=1.7*
Y
```



91

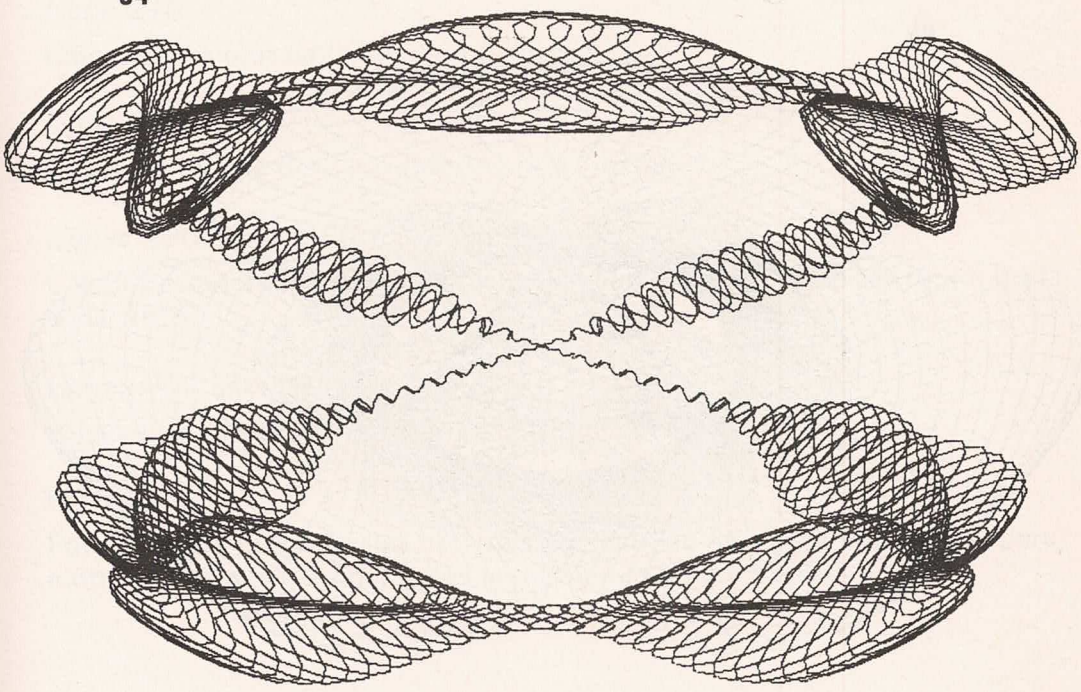


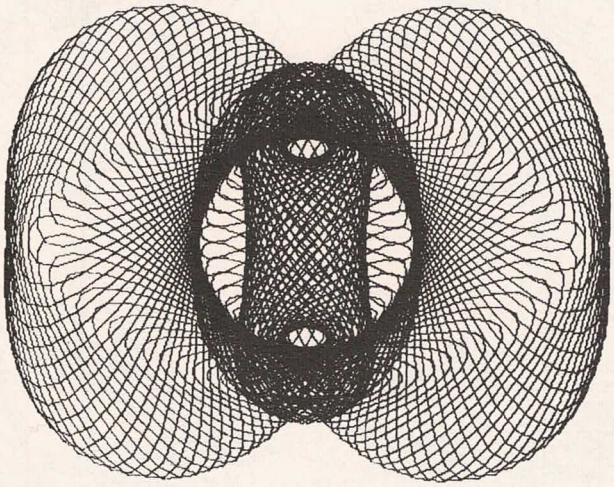
92



93

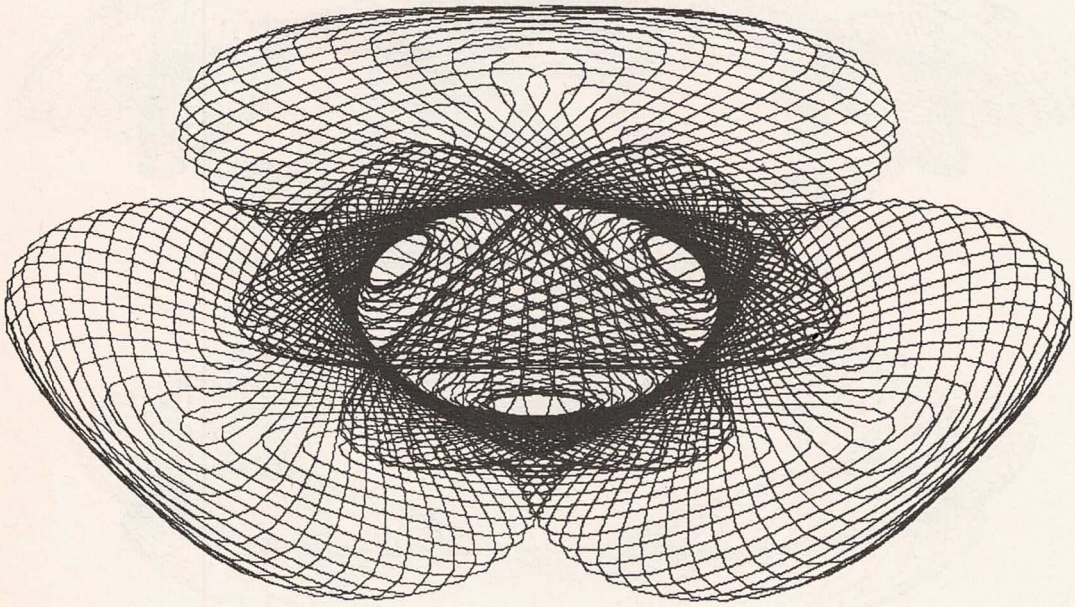
94





95

96



Le programme COURBES SPIRALES

Cette fois-ci, on imagine une planète décrivant une ellipse qui pivote autour de son centre (et non pas autour de l'un de ses foyers comme en mécanique réelle) et dont le rayon diminue.

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres :

N: nombre de segments élémentaires composant la courbe.

T: nombre de tours effectués par la planète.

R: paramètre d'aplatissement de l'ellipse (pour R = 1, on obtient un cercle).

L: facteur de diminution entre la première ellipse parcourue et la dernière.

Ligne 200

Début de la boucle principale. Comme dans les 2 programmes précédents.

Ligne 210

Calcul du facteur de diminution à l'instant I.

Ligne 220

Calcul de l'angle dont l'ellipse pivote à l'instant I.

Lignes 230, 240, 250, 260, 270

Calcul des coordonnées de la planète à l'instant I (formules de rotation à la ligne 250).

Lignes 300, 310, 400, 500

Analogues à celles du programme COURBES ORBITALES.

REMARQUE

Pour ces trois programmes, le temps d'exécution est directement lié au paramètre N : plus N est grand plus le dessin est long à obtenir.

Programme de base « COURBES SPIRALES »

```
10 'COURBES SPIRALES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=2000:T=40:R=.8:L=.1
200 FOR I=0 TO N
210 RR=L^(I/N)
220 AN=2*PI*I/N
230 X=RR*COS(T*AN):Y=RR*R*SIN(T*AN)
240 CO=COS(AN):SI=SIN(AN)
250 XX=X*CO-Y*SI:YY=X*SI+Y*CO
260 X%=INT(NP/2*(1+XX))
270 Y%=INT(NP/2*(1+YY))
300 IF I=0 THEN LPRINT"M";X%;",";Y%
310 IF I>0 THEN LPRINT"D";X%;",";Y%
400 NEXT I
500 END
```

Modifications pour le dessin 97

```
10 'COURBES SPIRALES
```

Modifications pour le dessin 98

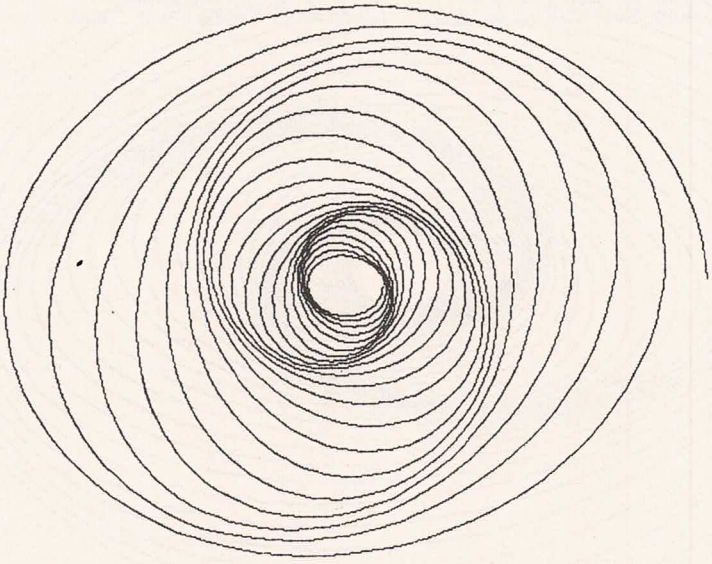
```
10 'COURBES SPIRALES
100 N=3000:T=60:R=.1:L=.1
270 Y%=INT(1.3*NP/2*(1+YY))
```

Modifications pour le dessin 99

```
10 'COURBES SPIRALES
100 N=2000:T=40:R=.8:L=.1
270 Y%=INT(1.7*NP/2*(1+YY))
```

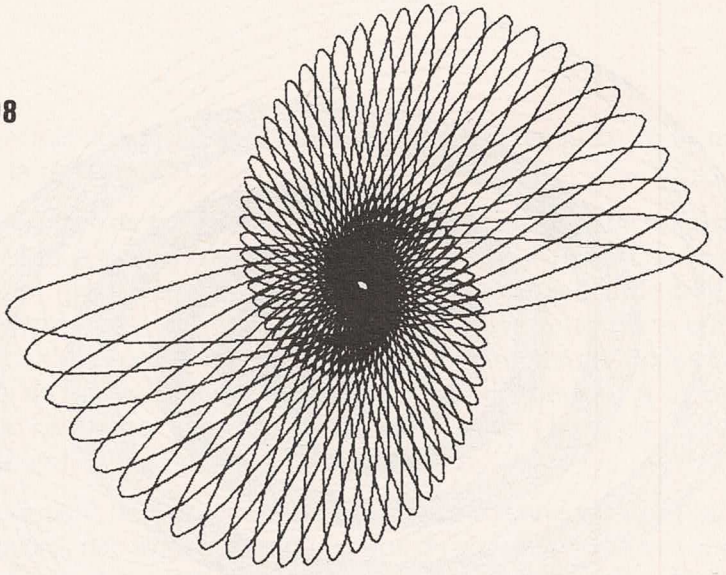
Modifications pour le dessin 100

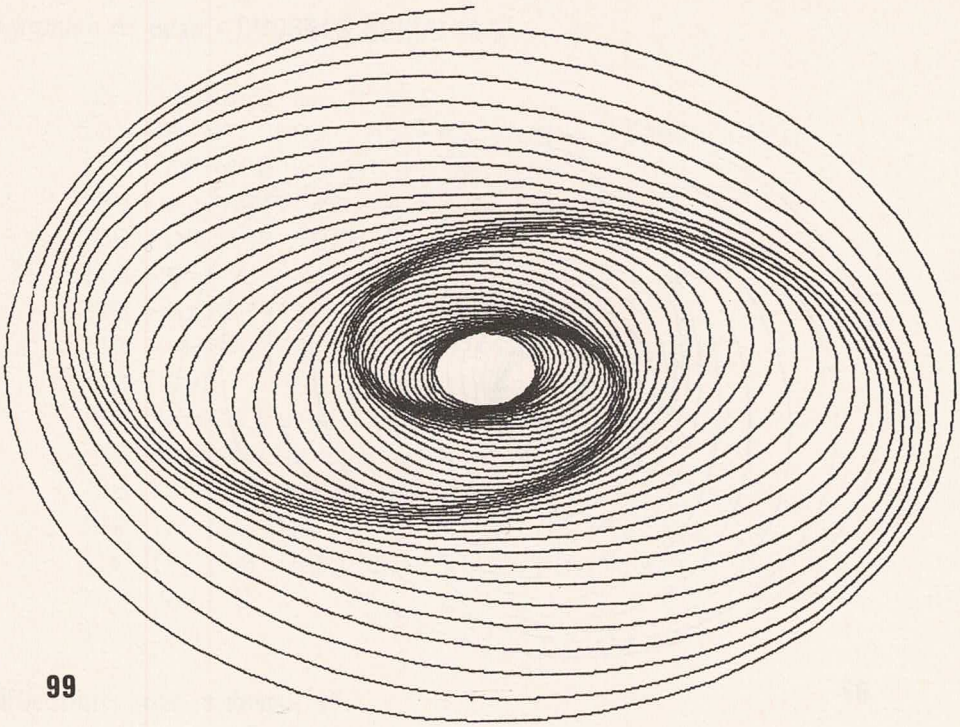
```
10 'COURBES SPIRALES
100 N=9000:T=50:R=.7:L=.05
220 AN=8*PI*I/N
270 Y%=INT(1.4*NP/2*(1+YY))
```



97

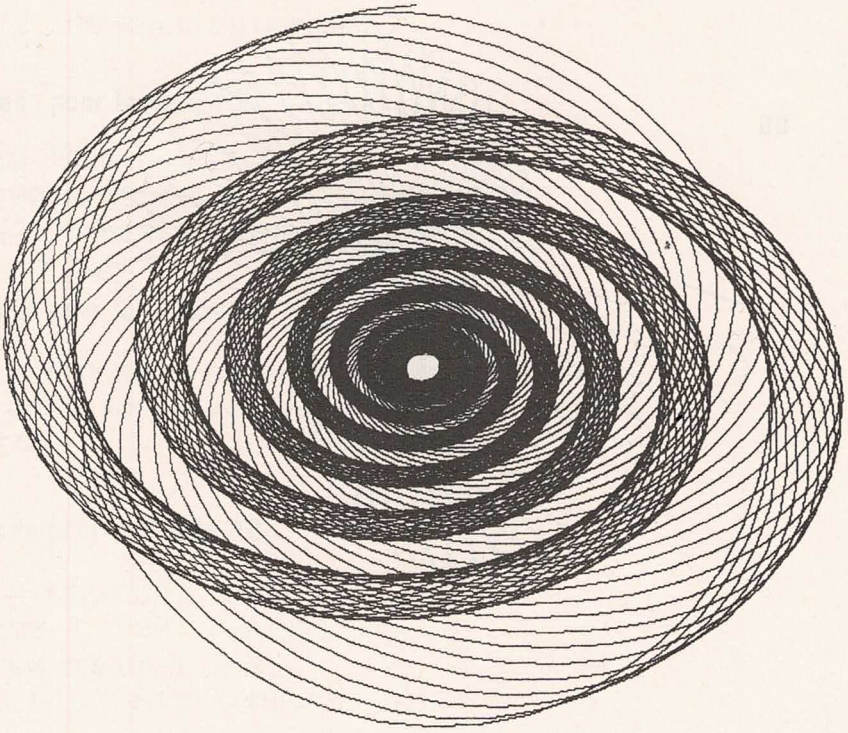
98





99

100



6 Dessins linéaires

On appelle parfois dessin linéaire un dessin composé uniquement de segments de droites (voir la référence [12]).

Puisque le seul ordre de tracé que nous utilisons est un ordre de tracé de segment, ou pourrait dire que tous les dessins proposés dans ce livre sont des dessins linéaires. Cependant, pour beaucoup d'entre eux, nous utilisons de très petits segments quasiment imperceptibles, qui, mis bout à bout, dessinent une courbe ou une autre structure dont la décomposition en segments de droites n'est en fait qu'un artifice de représentation. On peut donc garder un sens à la notion de dessin linéaire en disant par exemple que c'est un dessin composé uniquement de segments de droites bien visibles.

Ce chapitre contient trois programmes de dessins linéaires qui nous feront constater que, là encore, des algorithmes d'une grande simplicité peuvent engendrer des dessins ayant une structure plus compliquée que ne le laissait prévoir le principe de construction. A nouveau l'ordinateur nous fait découvrir qu'une idée élémentaire appliquée avec persévérance et exactitude peut donner naissance à de jolies figures imprévues.

Le programme BIPARTI COMPLET

En théorie des graphes, on appelle « graphe biparti complet » un graphe obtenu en joignant tout point d'un ensemble E à tout point d'un ensemble F. Le programme BIPARTI COMPLET dessine un graphe biparti complet entre un ensemble E composé de N + 1 points (numérotés de 0 à N) régulièrement espacés sur un segment AB, et un ensemble F composé lui aussi de N + 1 points (numérotés de 0 à N) régulièrement espacés sur un segment CD.

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres :

N : comme indiqué ci-dessus.
XA, YA : coordonnées du point A.
XB, YB : coordonnées du point B.
XC, YC : coordonnées du point C.
XD, YD : coordonnées du point D.

Ligne 200

Début de la boucle principale. A chaque parcours de la boucle le point numéro I de l'ensemble E est joint à tous les points de l'ensemble F.

Lignes 210, 220, 230

Calcul des coordonnées du point numéro I de l'ensemble E.

Ligne 300

Début de la boucle secondaire. A chaque parcours de cette boucle, le point numéro I de l'ensemble E est joint au point numéro J de l'ensemble F.

Ligne 310

Positionnement de la plume sur le point numéro I de l'ensemble E.

Lignes 320, 330, 340

Calcul des coordonnées du point numéro J de l'ensemble F.

Ligne 350

Tracé du segment entre le point numéro I de l'ensemble E et le point numéro J de l'ensemble F.

Lignes 400, 500

Fin des boucles. Fin du programme.

Le dessin 101 correspond à des segments AB et CD parallèles et tels que ABCD soit un carré. D'autres dispositions de A, B, C, D donnent les dessins 102, 103, 104.

Programme de base « BIPARTI COMPLET »

```
10 'BIPARTI COMPLET
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=10:XA=0:YA=0:XB=0:YB=NP:XC=NP:YC=0
:XD=NP:YD=NP
200 FOR I=0 TO N
210 X1=(I*XA+(N-I)*XB)/N
220 Y1=(I*YA+(N-I)*YB)/N
230 X%=INT(X1):Y%=INT(Y1)
300 FOR J= 0 TO N
310 LPRINT"M";X%;",";Y%
320 X2=(J*XC+(N-J)*XD)/N
330 Y2=(J*YC+(N-J)*YD)/N
340 XX%=INT(X2):YY%=INT(Y2)
350 LPRINT"D";XX%;",";YY%
400 NEXT J:NEXT I
500 END
```

Modifications pour le dessin 101

```
10 'BIPARTI COMPLET
100 N=10:XA=0:YA=0:XB=0:YB=NP:XC=NP:YC=0
:XD=NP:YD=NP
```

Modifications pour le dessin 102

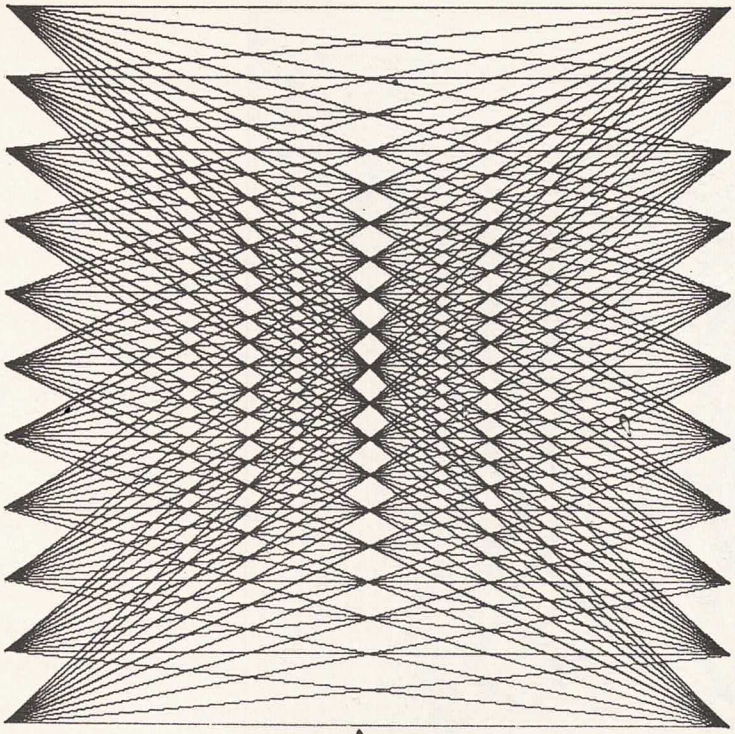
```
10 'BIPARTI COMPLET
100 N=15:XA=NP/2:YA=NP/15:XB=NP/2:YB=1.2
*NP:XC=0:YC=0:XD=NP:YD=0
```

Modifications pour le dessin 103

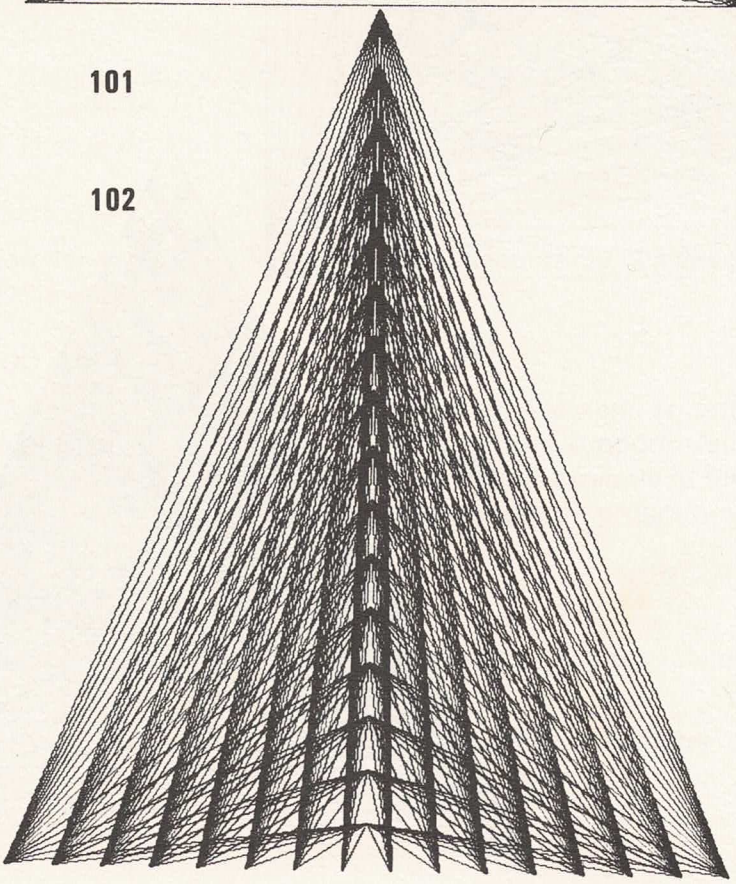
```
10 'BIPARTI COMPLET
100 N=19:XA=0:YA=0:XB=NP:YB=1.4*NP:XC=0:
YC=YB:XD=NP:YD=0
```

Modifications pour le dessin 104

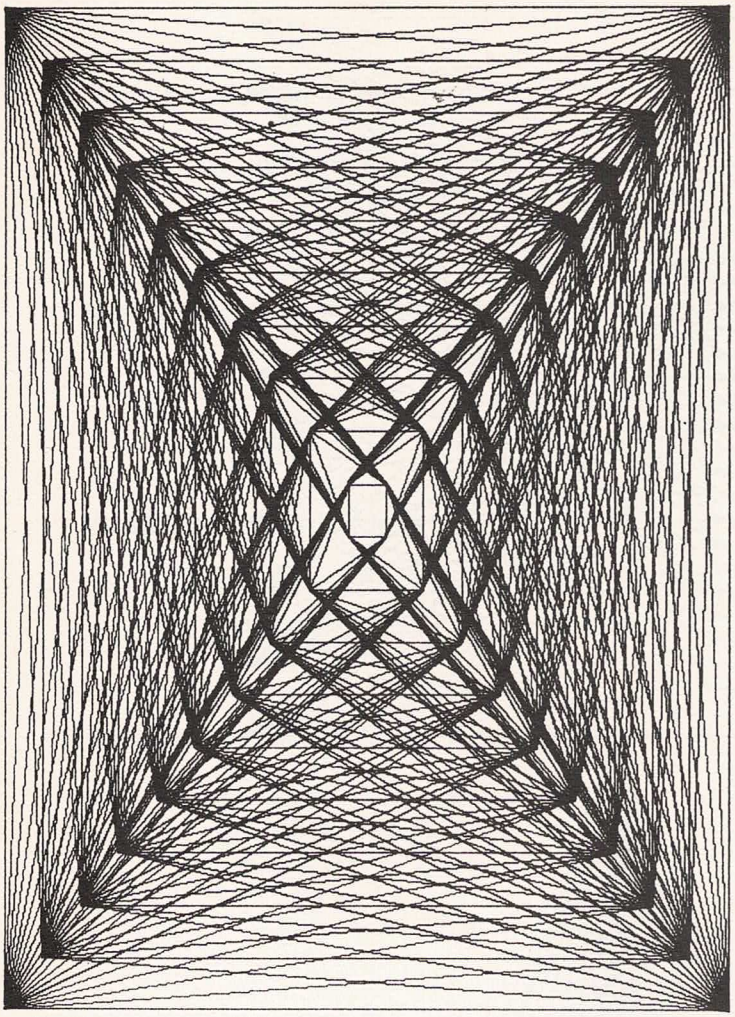
```
10 'BIPARTI COMPLET
100 N=16:XA=0:YA=0:XB=NP/2:YB=NP:XC=NP:Y
C=0:XD=NP/2:YD=NP
```



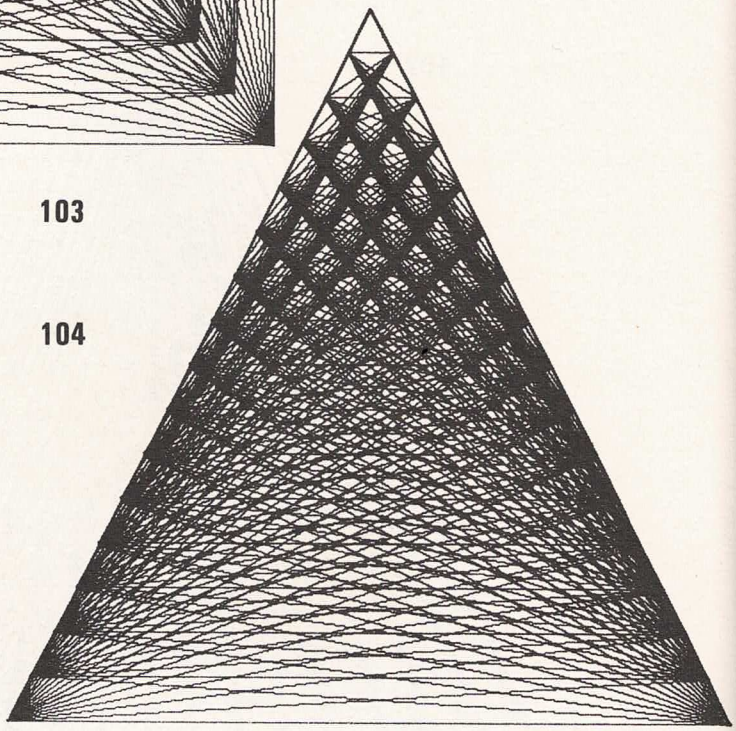
101



102



103



104

Le programme LINÉAIRES MODULO

On place N + 1 points sur une courbe de coefficients K1, K2 (analogue à celles utilisées au chapitre précédent) en numérotant ces points de 0 à N. Puis on joint le point numéro 0 au point numéro 0, le point numéro 1 au point numéro 2, le point numéro 2 au point numéro 4, le point numéro 3 au point numéro 6 etc. On dessine de cette façon M + 1 segments (en convenant que lorsqu'un numéro atteint N, on repart à zéro).

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres :

N + 1 : nombre de points placés sur la courbe support.

M + 1 : nombre de segments tracés.

K1, K2 : paramètres de la courbe support.

H : nombre de sauts pour l'extrémité terminale des segments tracés. Pour H = 2, on joint 0, 0 puis 1, 2, puis 2, 4, puis 3, 6, etc. Pour H = 3 on joint 0, 0, puis 1, 3, puis 2, 6, puis 3, 9, etc.

Ligne 110

Déclaration des tableaux qui contiendront les coordonnées des points placés sur la courbe support.

Lignes 200 à 230

Première boucle du programme. Pendant le déroulement de cette boucle, on calcule les coordonnées des points de la courbe support. Aucun tracé ne s'effectue pendant tous ces calculs, c'est ce qui explique le temps de quelques secondes entre le lancement de l'exécution du programme et le début du tracé.

Lignes 300 à 400

Deuxième boucle du programme. Réalisation du tracé selon le principe expliqué.

Si votre BASIC n'a pas la fonction MOD consulter le chapitre 3.

Programme de base « LINÉAIRES MODULO »

```
10 'LINEAIRES MODULO
50 LPRINT CHR$(18):LPRINT"M0,-900":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=400:M=N:K1=4:K2=5:H=2
110 DIM X%(N),Y%(N)
200 FOR I=0 TO N
210 X%(I)=INT(NP*.5*(1+SIN(K1*I*PI/N)))
220 Y%(I)=INT(NP*.75*(1+COS(K2*I*PI/N)))
230 NEXT I
300 FOR I=0 TO M
310 I1=I MOD N:I2=H*I MOD N
320 LPRINT"M";X%(I1);",";Y%(I1)
330 LPRINT"D";X%(I2);",";Y%(I2)
400 NEXT I
500 END
```

Modifications pour le dessin 105

```
10 'LINEAIRES MODULO
100 N=400:M=N:K1=4:K2=5:H=2
```

Modifications pour le dessin 106

```
10 'LINEAIRES MODULO
100 N=500:M=N:K1=11/7:K2=7/3:H=3
```

Modifications pour le dessin 107

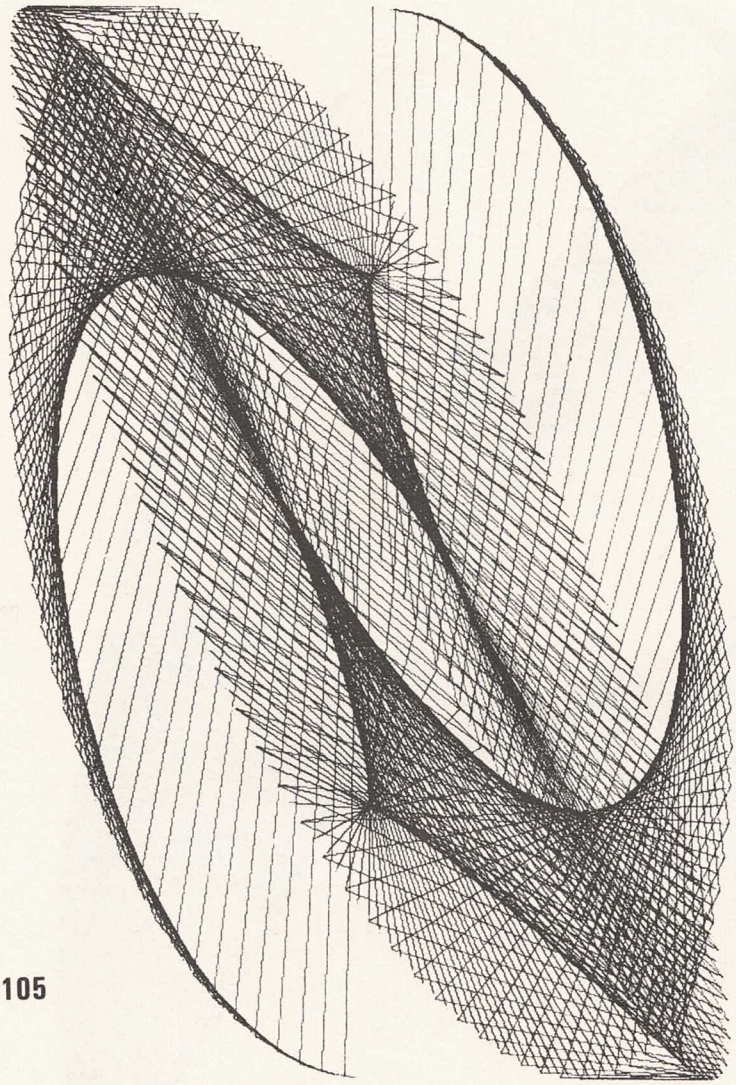
```
10 'LINEAIRES MODULO
100 N=400:M=N:K1=4:K2=2:H=2
```

Modifications pour le dessin 108

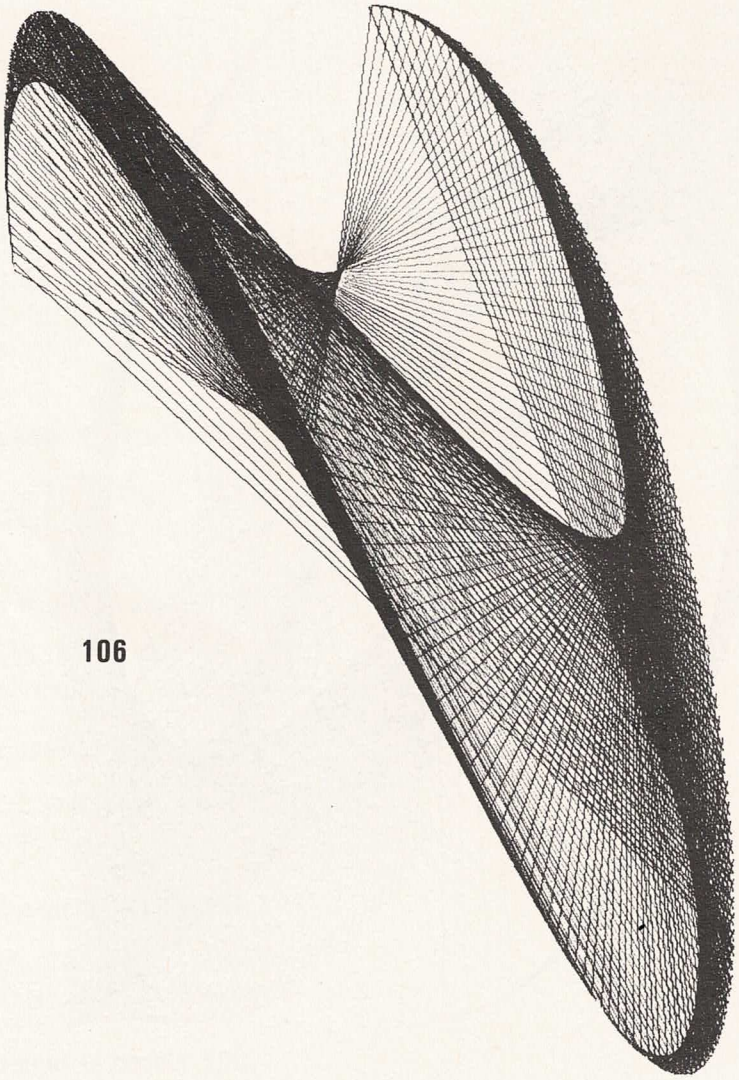
```
10 'LINEAIRES MODULO
100 N=300:M=N:K1=5:K2=3:H=2
```

Modifications pour le dessin 109

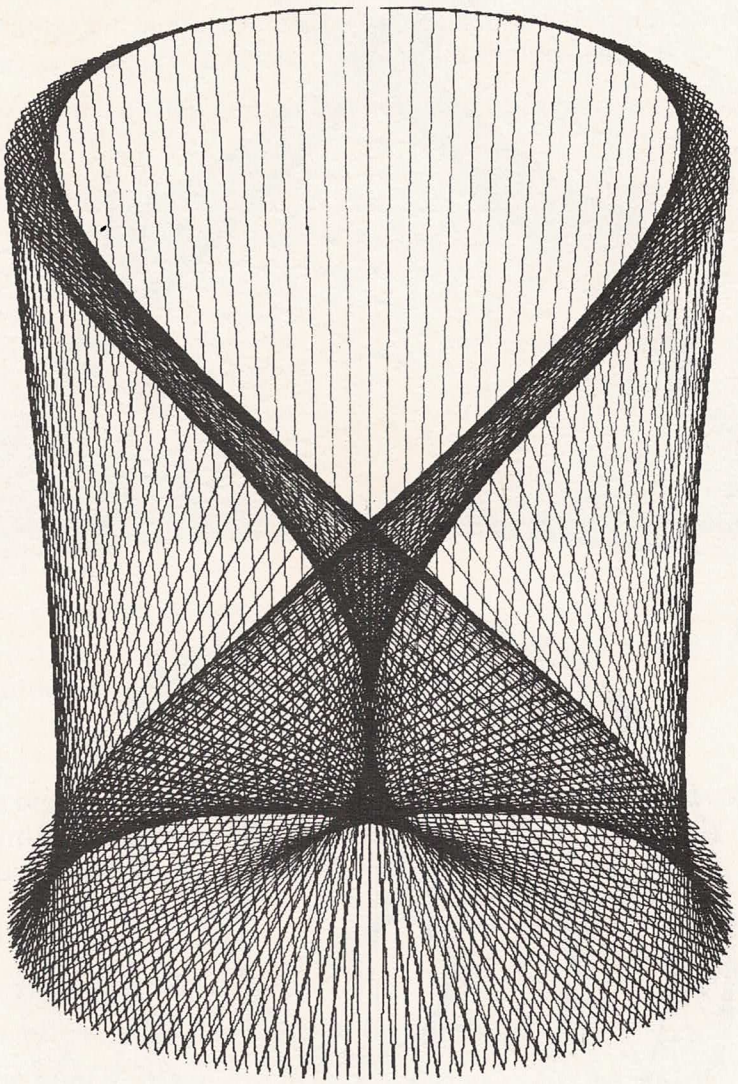
```
10 'LINEAIRES MODULO
100 N=400:M=200:K1=2:K2=2:H=9
220 Y%(I)=INT(NP*.5*(1+COS(K2*I*PI/N)))
310 I1=8*I MOD N:I2=H*I MOD N
```



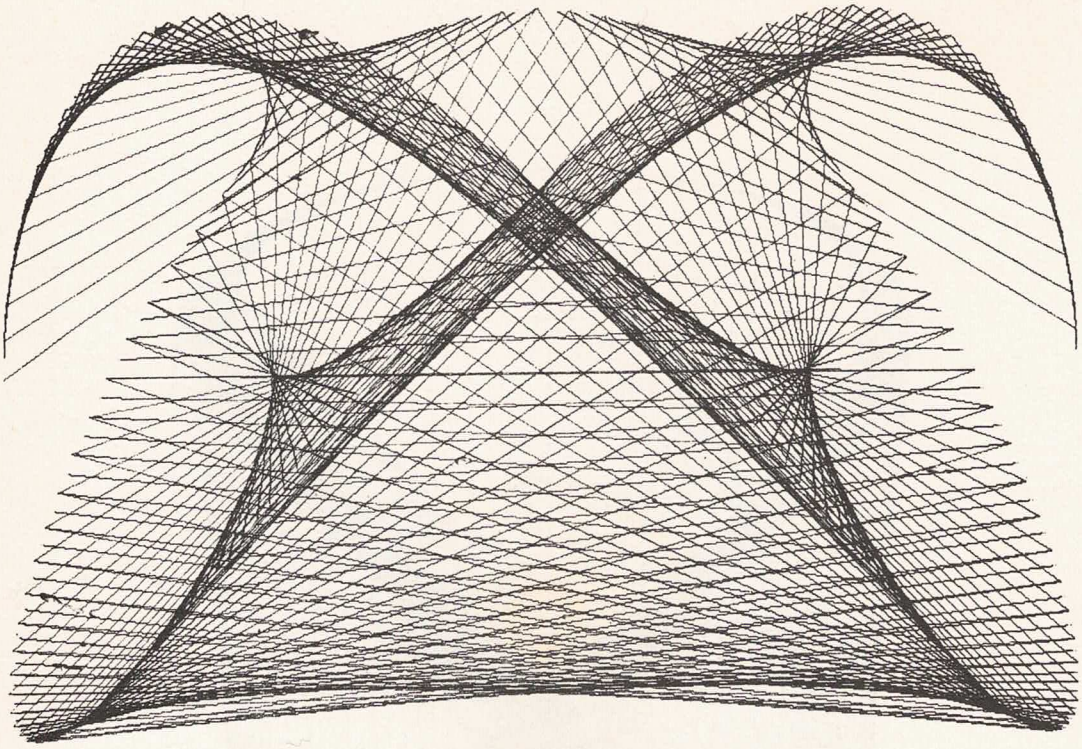
105



106

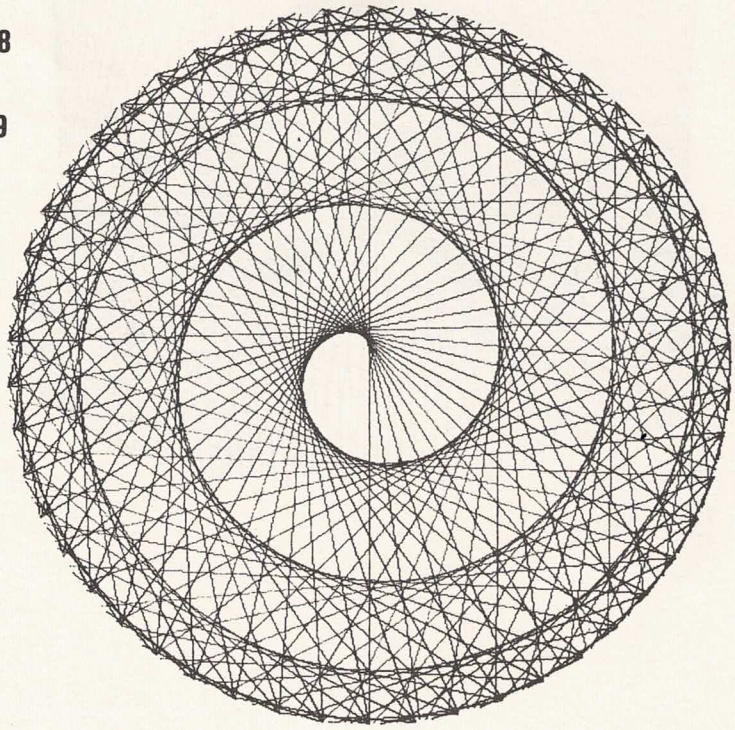


107



108

109



Le programme LINÉAIRES BATONS

Imaginons qu'on photographie à intervalles réguliers et rapprochés un bâton dont le centre décrit un cercle et qui en même temps tourne sur lui-même.

En superposant toutes les photos obtenues on obtiendra le dessin 110 ou le dessin 111.

Si maintenant, on aplatit une telle figure et qu'on la répète plusieurs fois à des tailles décroissantes on obtiendra des dessins comme les dessins 112, 113, 114.

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres :

N : nombre de photographies du bâton prises à chaque tour du bâton autour du point central.

M : nombre de séries de photos.

K : nombre de tours effectués par le bâton sur lui-même quand il parcourt un tour autour du point central.

Ligne 200

Début de la boucle principale. Chaque parcours de cette boucle correspond à une série entière de photographies. Pour les dessins 110 et 111 il n'y a qu'un seul parcours de cette boucle car M = 1.

Ligne 210

Fixation des paramètres :

R1 : rayon du cercle parcouru par le centre du bâton.

R2 : demi-longueur du bâton.

Lorsqu'on veut faire plusieurs séries de photographies (M>1), il faut que R1 et R2 dépendent de I comme pour les dessins 112, 113, 114.

Ligne 300

Début de la boucle secondaire. A chaque parcours de cette boucle un bâton est dessiné.

Ligne 310

Calcul de l'angle qui détermine la position du centre du bâton à l'instant J.

Lignes 320, 330, 340, 350

Calcul des coordonnées des extrémités du bâton à l'instant J.

Lignes 400, 410

Tracé du bâton.

Lignes 500, 1000

Fin des boucles. Fin du programme.

Programme de base « LINÉAIRES BÂTONS »

```
10 'LINEAIRES BATONS
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N=100:M=1:K=5
200 FOR I=1 TO M
210 R1=NP/4:R2=NP*5/24
300 FOR J=0 TO N-1
310 AN=2*J*PI/N
320 XD=NP/2+R1*COS(AN)+R2*COS(K*AN)
330 YD=NP/2+R1*SIN(AN)+R2*SIN(K*AN)
340 XA=NP/2+R1*COS(AN)+R2*COS(K*AN+PI)
350 YA=NP/2+R1*SIN(AN)+R2*SIN(K*AN+PI)
400 LPRINT"M";INT(XD);",";INT(YD)
410 LPRINT"D";INT(XA);",";INT(YA)
500 NEXT J:NEXT I
1000 END
```

Modifications pour le dessin 110

```
10 'LINEAIRES BATONS
```

Modifications pour le dessin 111

```
10 'LINEAIRES BATONS
100 N=600:M=1:K=5
210 R1=NP/4:R2=NP*5/24
```

Modifications pour le dessin 112

```
10 'LINEAIRES BATONS
100 N=100:M=6:K=6
210 R1=NP/3*.7^(I-1):R2=NP/8*.9^(I-1)
330 YD=NP/2+R1*SIN(AN)+R2*SIN(K*AN):YD=1
.2*YD
350 YA=NP/2+R1*SIN(AN)+R2*SIN(K*AN+PI):Y
A=1.2*YA
```

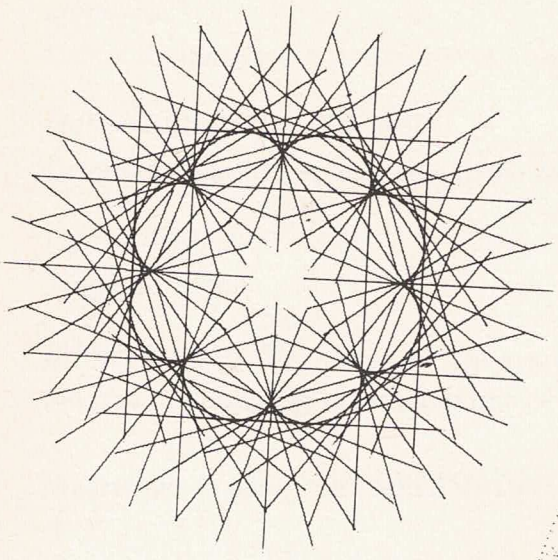
Modifications pour le dessin 113

```
10 'LINEAIRES BATONS
100 N=300:M=4:K=3
210 R1=NP/3*.6^(I-1):R2=NP/8*.6^(I-1)
```

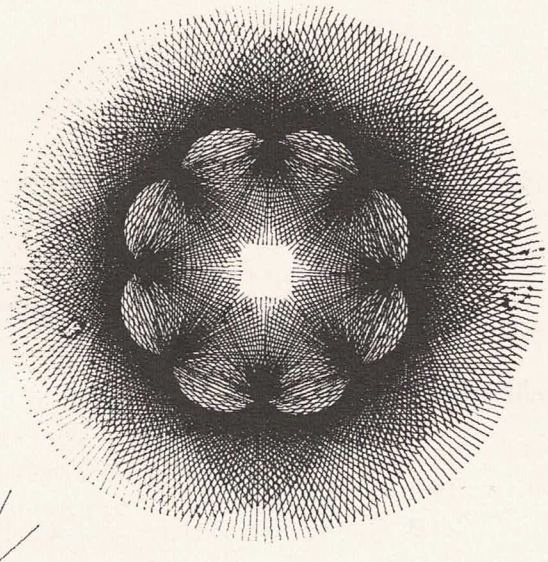
Modifications pour le dessin 114

```
10 'LINEAIRES BATONS
100 N=300:M=7:K=7
210 R1=NP/3*.8^(I-1):R2=NP/12*.8^(I-1)

330 YD=NP/2+R1*SIN(AN)+R2*SIN(K*AN):YD=1
.3*YD
350 YA=NP/2+R1*SIN(AN)+R2*SIN(K*AN+PI):Y
A=1.3*YD
```

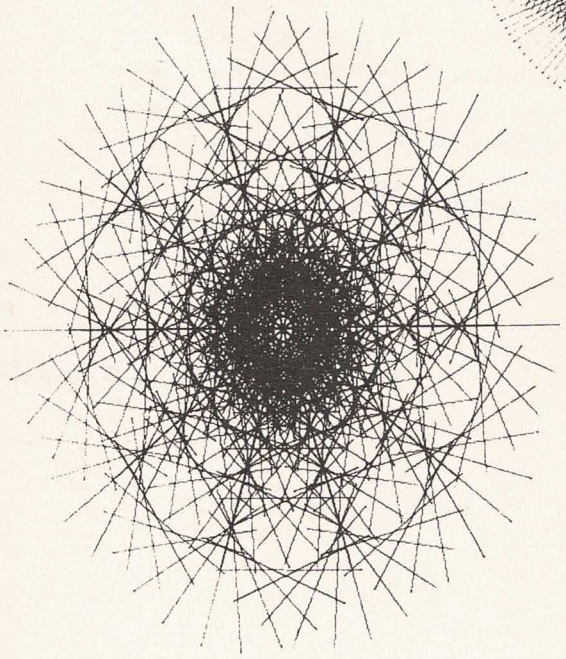


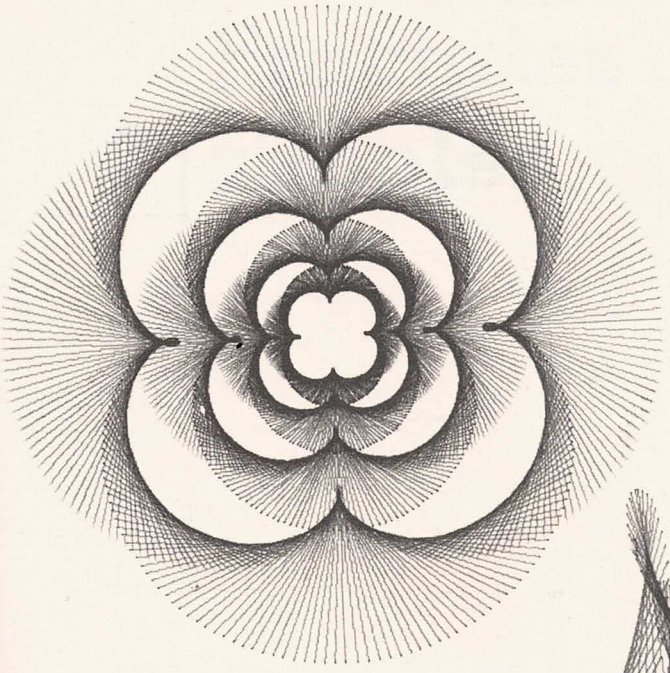
110



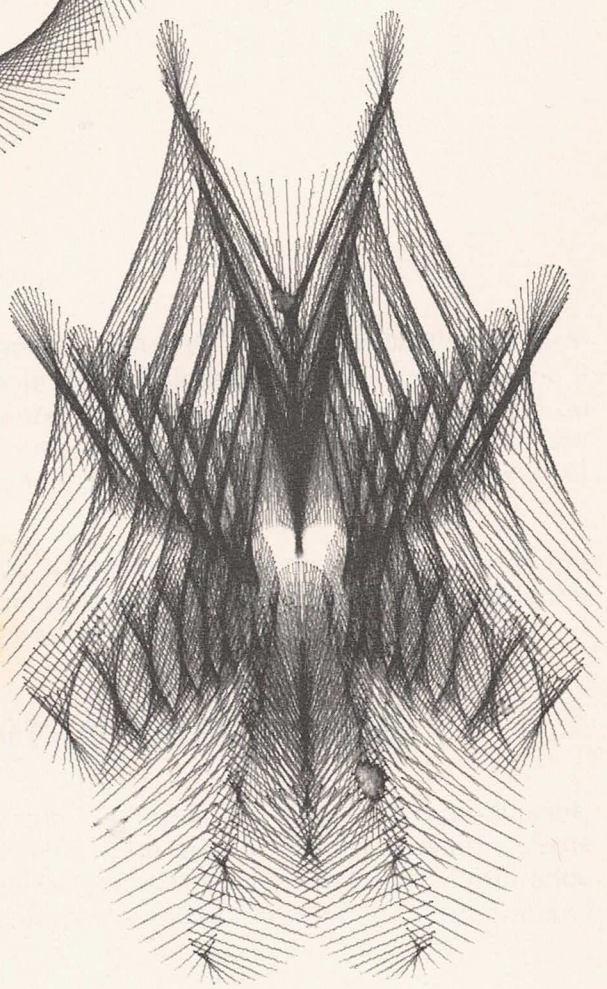
111

112





113



114

7 Fractales simples

Le premier programme de ce chapitre permet de tracer une famille très intéressante de courbes fractales (pour des explications sur le mot « fractale » voyez le chapitre 4) ; ce sont toutes les courbes basées sur le principe du flocon de Von Koch. Nous qualifions ces fractales de simples, car il existe diverses généralisations du principe utilisé (consulter les références [24] [25].

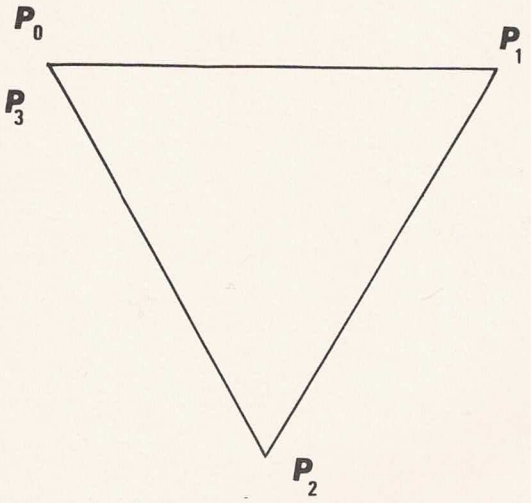
Les deux autres programmes sont construits à partir du premier. L'un d'eux permet d'arrondir les angles entre les segments successifs, l'autre déforme les figures ce qui donne des résultats assez extraordinaires.

Le programme FRACTALES SIMPLES

(Le texte de ce paragraphe ainsi que le programme lui-même, reproduisent, avec quelques modifications, le texte d'un article et un programme parus dans le numéro de mai 1984 de la revue « Sciences et Vie Micro », référence [6]. On trouvera des exemples complémentaires d'utilisation du programme dans la référence [7]).

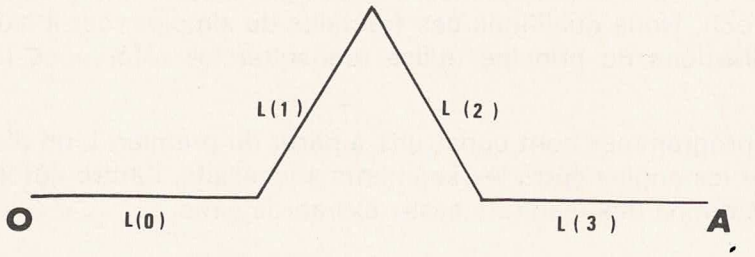
Nous allons maintenant expliquer le principe de construction des « fractales simples » sur l'exemple du fameux flocon de Von Koch (référence [34]) qui est une courbe n'admettant aucune tangente, et ayant une longueur infinie.

On part d'une famille de segments P_0, P_1, P_2, P_3 avec $P_0 = P_3$ et telle que les trois points P_0, P_1, P_2 forment un triangle équilatéral. On appelle cette suite de segments « courbe de départ », on la note CØ.



COURBE DE DEPART CØ

On remplace alors chaque segment de CØ par une suite de segments que nous appelons modèle :



MODELE

Ceci donne le dessin 115 que nous appelons courbe C1.

Sur le dessin 115, on effectue la même opération de substitution (chaque segment est remplacé par le modèle). On obtient le dessin 116 que nous appelons courbe C2.

Et on recommence, ce qui donne la courbe C3 (dessin 117) la courbe C4 (dessin 118) la courbe C5 (dessin 119).

La suite de courbes CØ, C1, C2,... « converge » vers une courbe limite qui est une courbe fractale, (ici le flocon de Von Koch). Quand nous disons que la suite de courbes converge, cela signifie qu'à partir d'un certain nombre d'étapes, il n'y a plus de différence sensible entre une courbe et la suivante (ce qui est visible sur les dessins).

Bien sûr, pour représenter des courbes fractales, nous nous contentons de dessiner l'une des courbes CØ, C1,...

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres :

M : nombre de segments de la courbe de départ.

N : nombre de segments du modèle.

K : numéro de la courbe que l'on veut obtenir
(K = 4 correspond au dessin 118).

Ligne 110

Déclaration des tableaux qui vont contenir :

— les coordonnées des points de la courbe de départ (tableaux X [M], Y [M])

— les longueurs des segments du modèle (tableau L [N—1])

— les angles des segments du modèle avec l'horizontale (tableau A [N—1]).

Lignes 120, 130

Fixation des coordonnées des points de la courbe de départ.

Ligne 140

Fixation des longueurs des segments du modèle. On prend comme unité de mesure la longueur du segment OA, donc : L (0) = L (1) = L (2) = L (3) =1/3.

Ligne 150

Fixation des angles des segments du modèle avec l'horizontale. Pour notre exemple A (0) = A (3) = 0, A (1) = — A (2) = PI/3.

Ligne 200

Début de la boucle principale. A chaque parcours de la boucle, toute la partie du dessin correspondant à l'un des segments de la courbe de départ se réalise.

Ligne 210

Calcul des coordonnées du point de départ D et du point d'arrivée A du segment numéro II de la courbe de départ.

Lignes 250 à 290

Initialisation des calculs et positionnement de la plume.

Lignes 300 à 500

Début de la boucle secondaire. A chaque parcours de cette boucle, un segment élémentaire est dessiné. Le principe des calculs est le suivant :

On imagine que chaque segment à dessiner entre D et A est numéroté. Il y a N ^ K segments.

Pour tracer le segment numéro I, on décompose I dans la base de numération N, on obtient $I = a_1\, a_2 \ldots a_r$, ceci permet de connaître sa longueur :

$LL = L\emptyset * L(a_1) * L(a_2) * \ldots * L(a_r)$

et son angle avec l'horizontale :

$AA = A\emptyset + A(a_1) + A(a_2) + \ldots + A(a_r)$

Tous ces calculs se font dans la boucle 400-450.

Une fois connus LL et AA, on trace sans peine le segment numéro I (lignes 470-490).

Lignes 500, 600, 1000

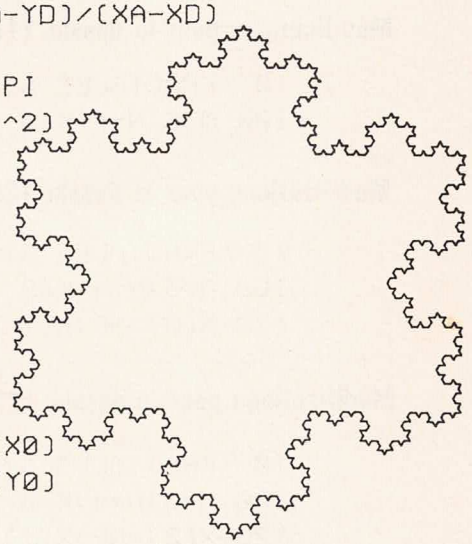
Fin des boucles. Fin du programme.

Les dessins 120, 121, 122, 123 représentent les étapes successives de la construction de ce qu'on appelle l'antiflocon de Von Koch : la croissancede la courbe de départ se fait vers l'intérieur au lieu de se faire vers l'extérieur.

Les dessins 124-131 constituent les étapes de la construction d'une fractale qui se recoupe de façon incompréhensible mais malgré tout régulière.

Programme de base « FRACTALES SIMPLES »

```
10 'FRACTALES SIMPLES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 M=3:N=4:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=NP*.5:X(3)=0
130 Y(0)=SQR(3)/2*NP:Y(1)=Y(0):Y(2)=0:Y(
3)=Y(0)
140 L(0)=1/3:L(1)=L(0):L(2)=L(0):L(3)=L(
0)
150 A(0)=0:A(1)=PI/3:A(2)=-A(1):A(3)=0
200 FOR II=0 TO M-1
210 XD=X(II):YD=Y(II):XA=X(II+1):YA=Y(II
+1)
250 X0=XD:Y0=YD
260 X1=INT(X0):Y1=INT(Y0)
270 LPRINT "M";X1;",";Y1
280 IFXA<>XD THEN A0=ATN((YA-YD)/(XA-XD)
) ELSE A0=PI/2*SGN(YA-YD)
285 IF (XA-XD)<0 THEN A0=A0+PI
290 L0=SQR((XA-XD)^2+(YA-YD)^2)
300 FOR I=0 TO N^K-1
310 LL=L0:AA=A0:T1=I
390 IF K=0 THEN 470
400 FOR J=K-1 TO 0 STEP -1
410 R=N^J:T2=INT(T1/R)
430 AA=AA+A(T2):LL=LL*L(T2)
440 T1=T1-T2*R
450 NEXT J
470 X0=X0+LL*COS(AA):X1=INT(X0)
480 Y0=Y0+LL*SIN(AA):Y1=INT(Y0)
490 LPRINT"D";X1;",";Y1
500 NEXT I
600 NEXT II
1000 END
```



Modifications pour le dessin 115

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=1
```

Modifications pour le dessin 116

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=2
```

Modifications pour le dessin 117

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=3
```

Modifications pour le dessin 118

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=4
```

Modifications pour le dessin 119

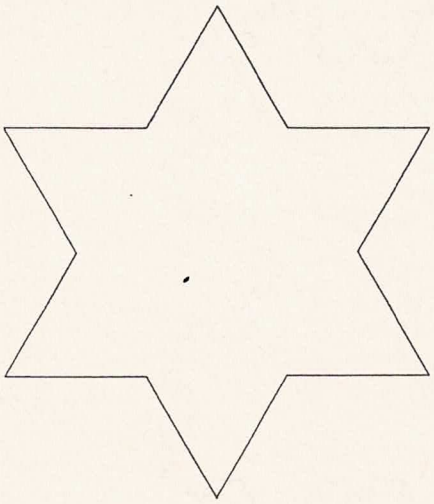
```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=5
```

Modifications pour le dessin 120

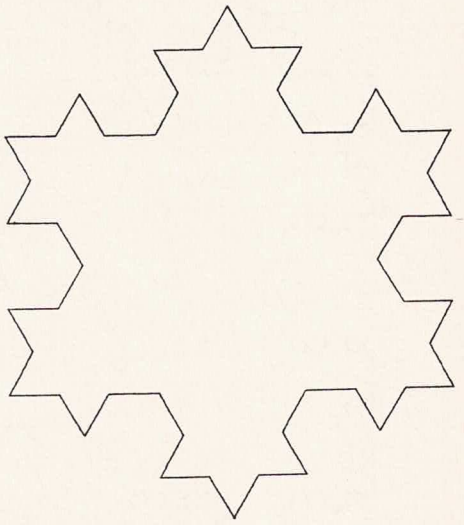
```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=2
120 X(0)=NP:X(1)=0:X(2)=NP*.5:X(3)=NP
```

Modifications pour le dessin 121

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=3
120 X(0)=NP:X(1)=0:X(2)=NP*.5:X(3)=NP
```

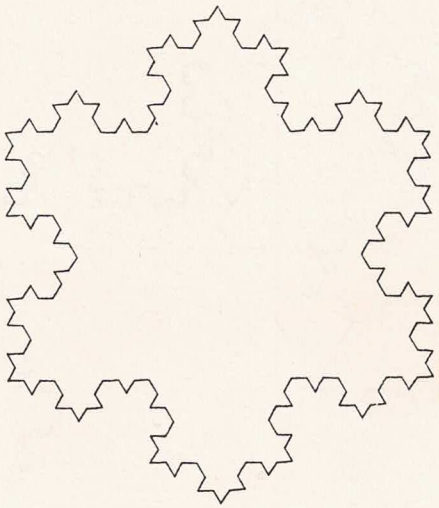


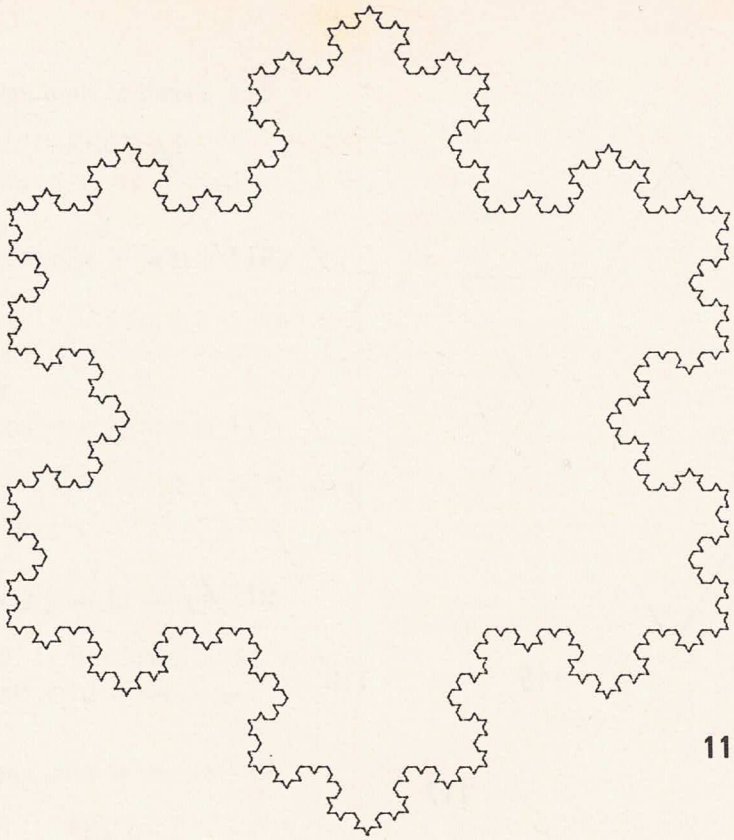
115



116

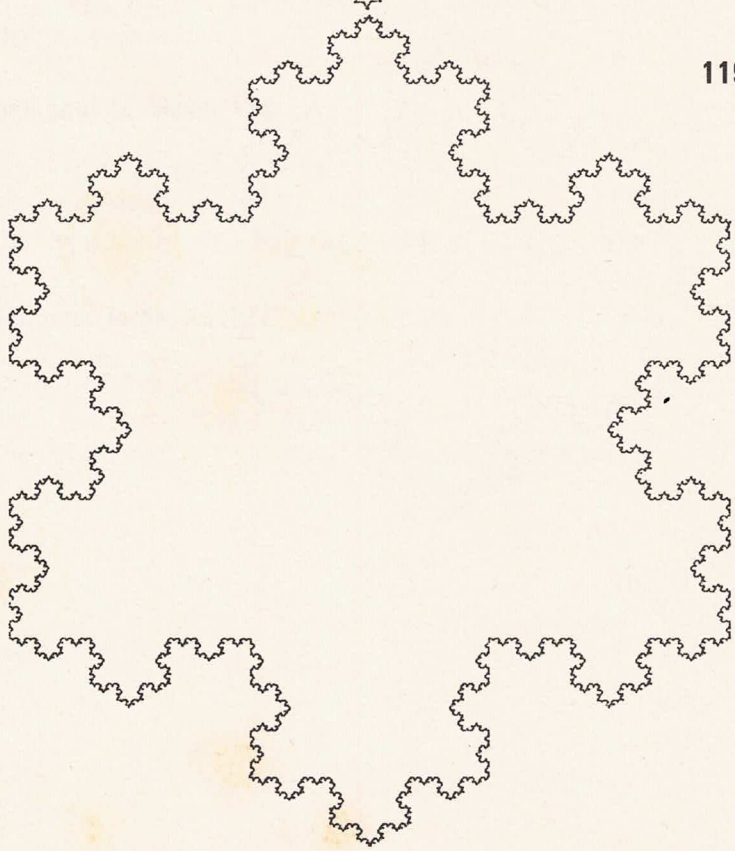
117

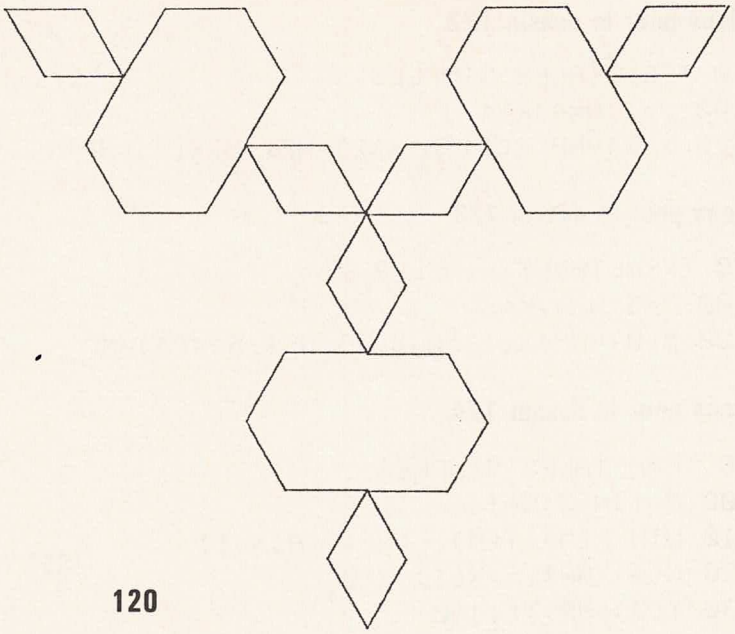




118

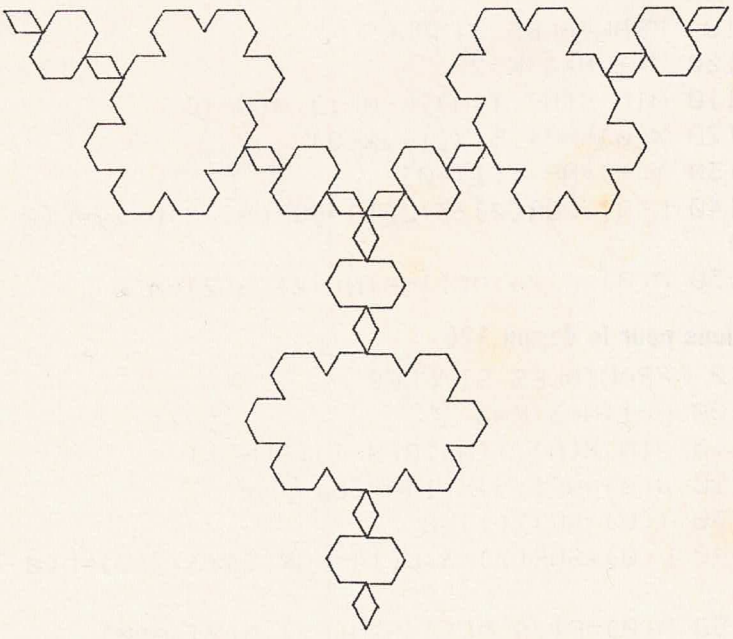
119





120

121



Modifications pour le dessin 122

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=4
120 X(0)=NP:X(1)=0:X(2)=NP*.5:X(3)=NP
```

Modifications pour le dessin 123

```
10 'FRACTALES SIMPLES
100 M=3:N=4:K=5
120 X(0)=NP:X(1)=0:X(2)=NP*.5:X(3)=NP
```

Modifications pour le dessin 124

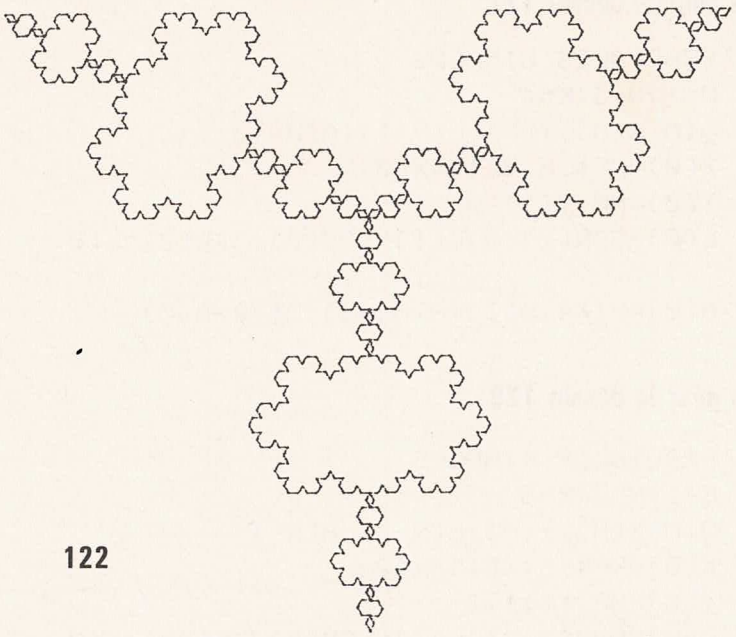
```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=1
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```

Modifications pour le dessin 125

```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=2
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```

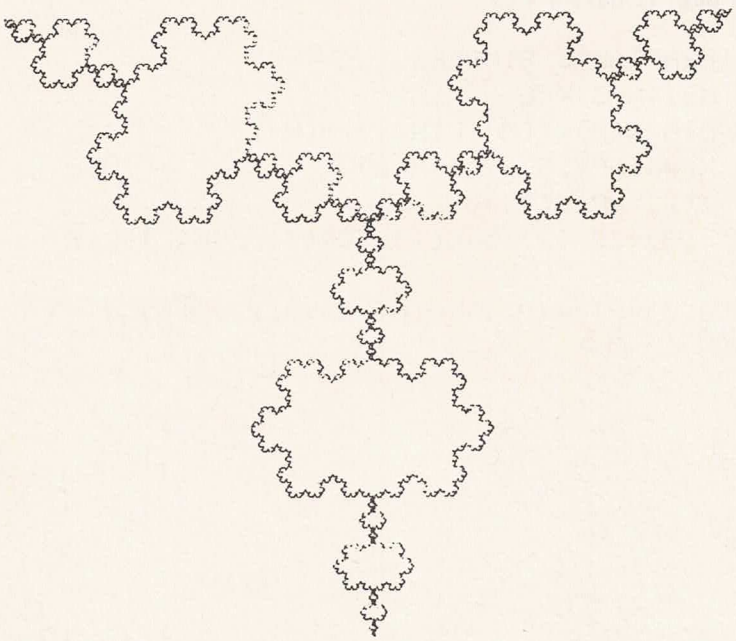
Modifications pour le dessin 126

```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=3
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```



122

123



Modifications pour le dessin 127

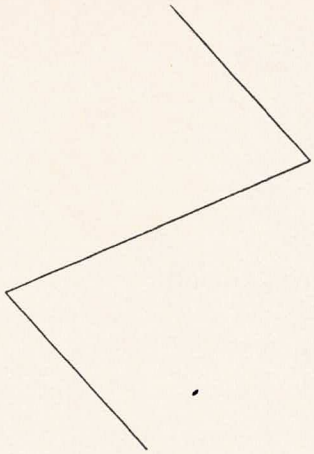
```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```

Modifications pour le dessin 128

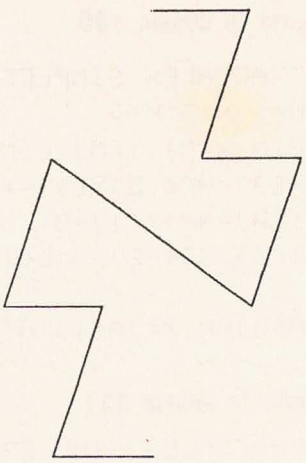
```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=5
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```

Modifications pour le dessin 129

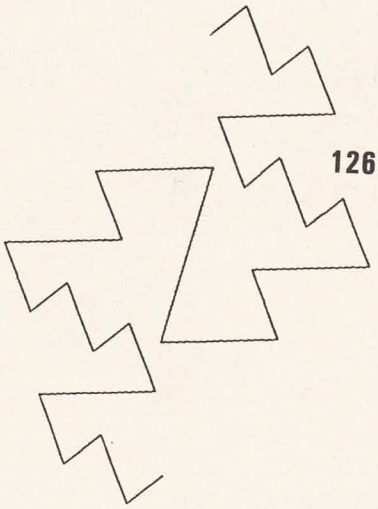
```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=6
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```



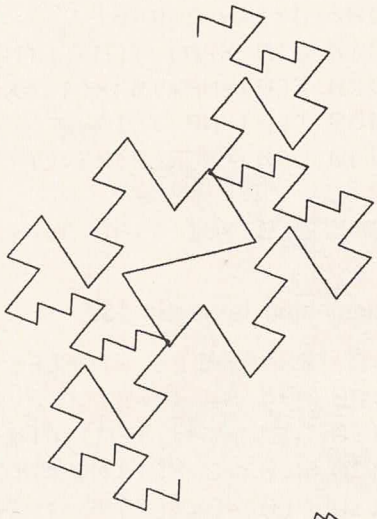
124



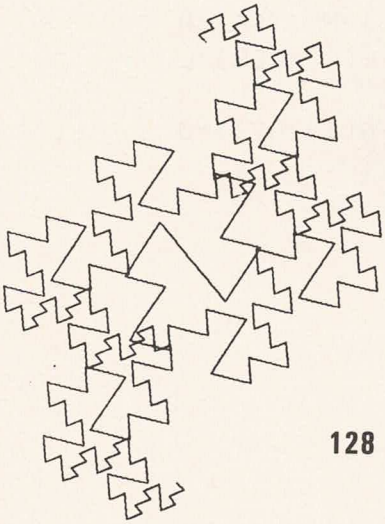
125



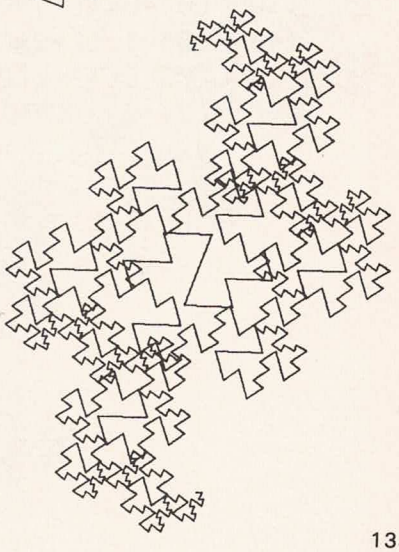
126



127



128



129

Modifications pour le dessin 130

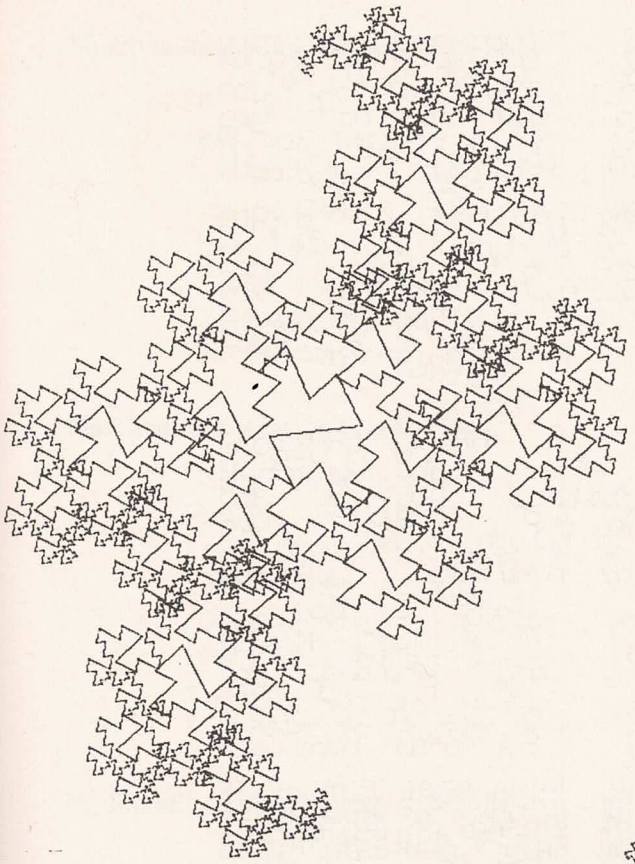
```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=7
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```

Modifications pour le dessin 131

```
10 'FRACTALES SIMPLES
100 M=1:N=3:K=8
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP*.5:X(1)=X(0)
130 Y(0)=NP:Y(1)=0
140 L(0)=SQR(2)/3:L(1)=SQR(5)/3:L(2)=L(0
)
150 A(0)=PI/4:A(1)=ATN(-2):A(2)=A(0)
```

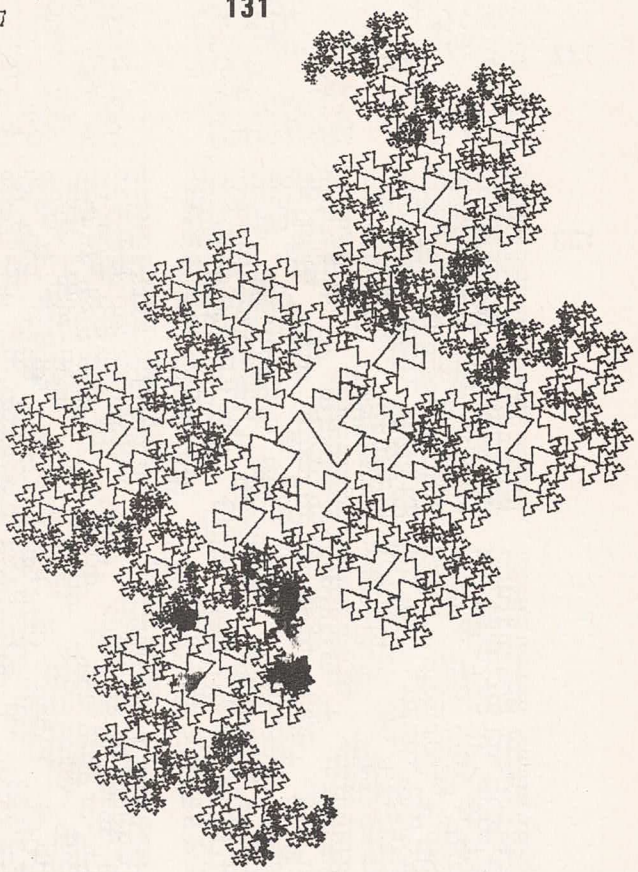
Modifications pour le dessin 132

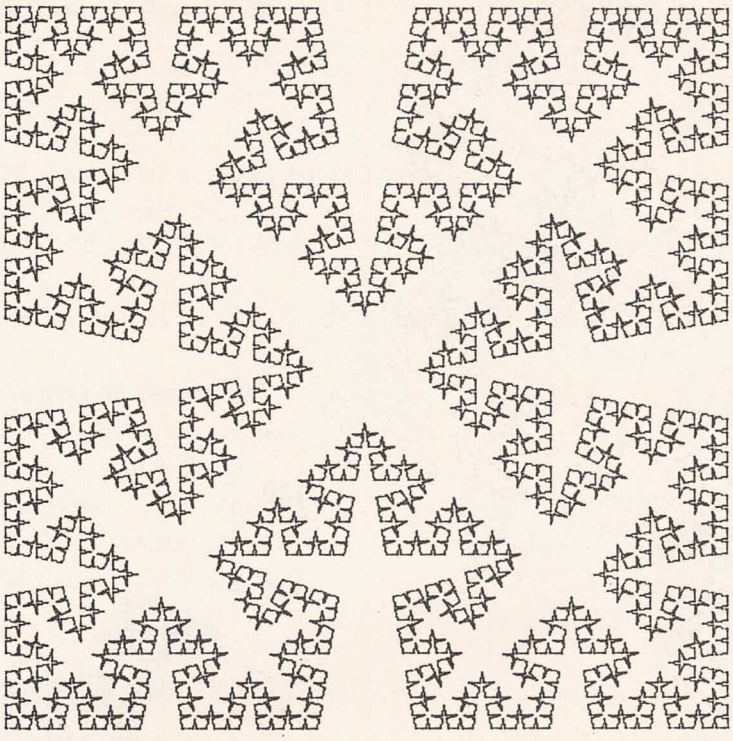
```
10 'FRACTALES SIMPLES
100 M=4:N=4:K=5
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=NP:X(3)=0:X(4)=0
130 Y(0)=0:Y(1)=0:Y(2)=NP:Y(3)=NP:Y(4)=0
140 L(0)=1/(2+2*COS(.45*PI)):L(1)=L(0):L
(2)=L(0):L(3)=L(0)
150 A(0)=0:A(1)=.45*PI:A(2)=-A(1):A(3)=0
```



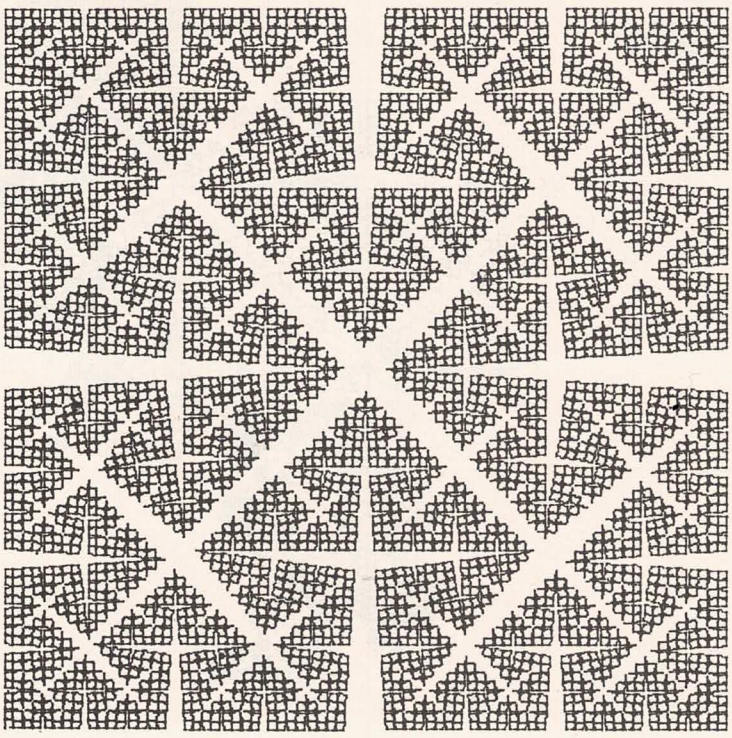
130

131





132



133

Modifications pour le dessin 133

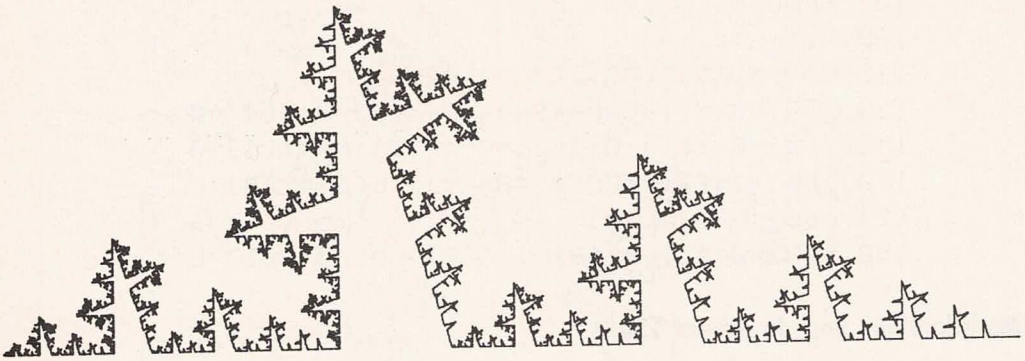
```
10 'FRACTALES SIMPLES
100 M=4:N=4:K=6
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=NP:X(3)=0:X(4)=0
130 Y(0)=0:Y(1)=0:Y(2)=NP:Y(3)=NP:Y(4)=0
140 L(0)=1/(2+2*COS(.48*PI)):L(1)=L(0):L
(2)=L(0):L(3)=L(0)
150 A(0)=0:A(1)=.48*PI:A(2)=-A(1):A(3)=0
```

Modifications pour le dessin 134

```
10 'FRACTALES SIMPLES
100 M=1:N=4:K=6
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=0
130 Y(0)=NP:Y(1)=-NP
140 L(0)=1/3:L(1)=L(0):L(2)=SQR(10)/9:L(
3)=5/9
150 A(0)=0:A(1)=PI/2:A(2)=-ATN(3):A(3)=0
```

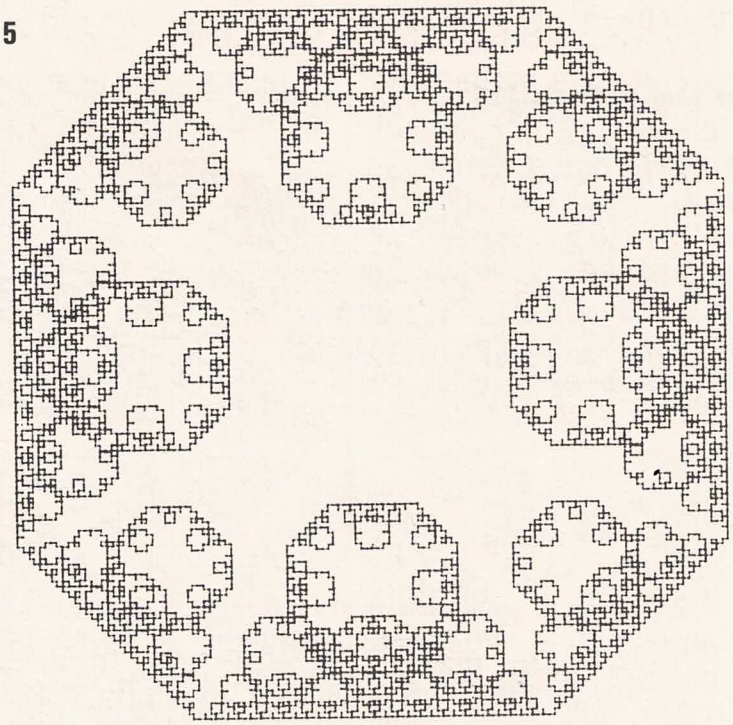
Modifications pour le dessin 135

```
10 'FRACTALES SIMPLES
100 M=2:N=2:K=12
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=NP/2:X(1)=X(0):X(2)=X(0)
130 Y(0)=NP*.25:Y(1)=NP*.75:Y(2)=NP*.25
140 L(0)=1/SQR(2):L(1)=L(0)
150 A(0)=PI/4:A(1)=-PI/4
```



134

135



Le programme FRACTALES SIMPLES ARRONDIES

Dans ce programme basé sur le précédent nous avons remplacé le tracé des segments par le tracé d'arcs d'ellipses qui se recollent.

Pour cela, nous avons modifié les lignes 100, 250, 260, 470, 480, 490 de manière à garder les coordonnées des trois derniers points qui permettent au sous-programme (placé des lignes 5 000 à 5 100) de tracer l'arc d'ellipse voulu.

Le paramètre supplémentaire S introduit à la ligne 100 détermine la finesse du tracé de l'arc d'ellipse qui est réalisé par un nombre fini, S + 1, de segments.

Programme de base « FRACTALES SIMPLES ARRONDIES »

```
10 'FRACTALES SIMPLES ARRONDIES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 M=1:N=7:K=2:S=5
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=0
130 Y(0)=NP:Y(1)=-NP
140 L(0)=1/2:L(1)=1/4:L(2)=1/4:L(3)=1/4:
L(4)=1/4:L(5)=1/2:L(6)=1/2
150 A(0)=0:A(1)=PI/2:A(2)=-PI:A(3)=0:A(4
)=PI/2:A(5)=-PI/2:A(6)=0
200 FOR II=0 TO M-1
210 XD=X(II):YD=Y(II):XA=X(II+1):YA=Y(II
+1)
```

```
250 X0=XD:X1=X0:X2=X0:Y0=YD:Y1=Y0:Y2=Y0
280 IFXA<>XD THEN A0=ATN((YA-YD)/(XA-XD)
) ELSE A0=PI/2*SGN(YA-YD)
285 IF (XA-XD)<0 THEN A0=A0+PI
290 L0=SQR((XA-XD)^2+(YA-YD)^2)
300 FOR I=0 TO N^K-1
310 LL=L0:AA=A0:T1=I
390 IF K=0 THEN 470
400 FOR J=K-1 TO 0 STEP -1
410 R=N^J:T2=INT(T1/R)
430 AA=AA+A(T2):LL=LL*L(T2)
440 T1=T1-T2*R
450 NEXT J
470 X0=X1:X1=X2:X2=X2+LL*COS(AA)
480 Y0=Y1:Y1=Y2:Y2=Y2+LL*SIN(AA)
490 GOSUB 5000
500 NEXT I
600 NEXT II
1000 END
5000 'SOUS-PROGRAMME POUR ARRONDIR
5010 UX=X1-X0:UY=Y1-Y0:WX=X2-X1:WY=Y2-Y1
5020 FOR K4=0 TO S
5030 AN=PI/2*K4/S:CO=COS(AN):SI=SIN(AN)
5040 XQ=(X0+X2+CO*(-WX)+SI*UX)/2
5050 YQ=(Y0+Y2+CO*(-WY)+SI*UY)/2
5060 IF I=0 THEN LPRINT"M";INT(XQ);",";I
NT(YQ)
5070 IF I>0 THEN LPRINT"D";INT(XQ);",";I
NT(YQ)
5080 NEXT K4
5100 RETURN
```

Modifications pour le dessin 136

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=7:K=1:S=20
```

Modifications pour le dessin 137

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=7:K=2:S=10
```

Modifications pour le dessin 138

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=7:K=3:S=5
```

Modifications pour le dessin 139

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=7:K=4:S=5
```

Modifications pour le dessin 140

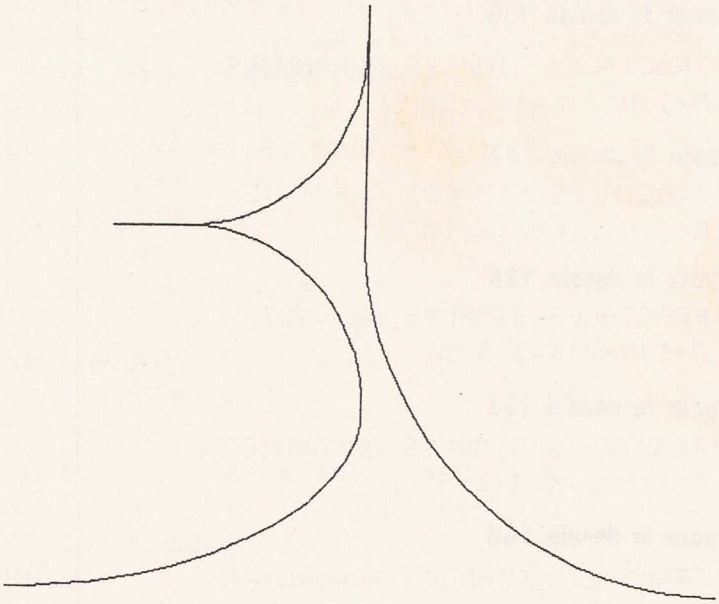
```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=4:K=1:S=10
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.5:L(1)=.45:L(2)=.45:L(3)=.5
150 A(0)=0:A(1)=PI/2:A(2)=-PI/2:A(3)=0
```

Modifications pour le dessin 141

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=4:K=2:S=5
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.5:L(1)=.45:L(2)=.45:L(3)=.5
150 A(0)=0:A(1)=PI/2:A(2)=-PI/2:A(3)=0
```

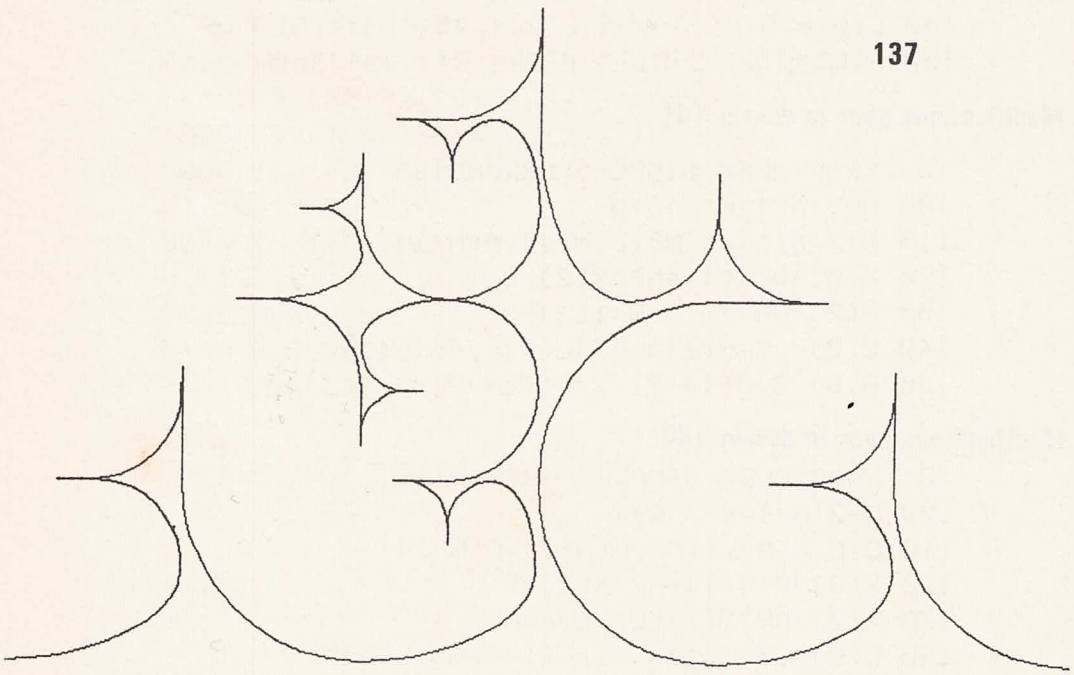
Modifications pour le dessin 142

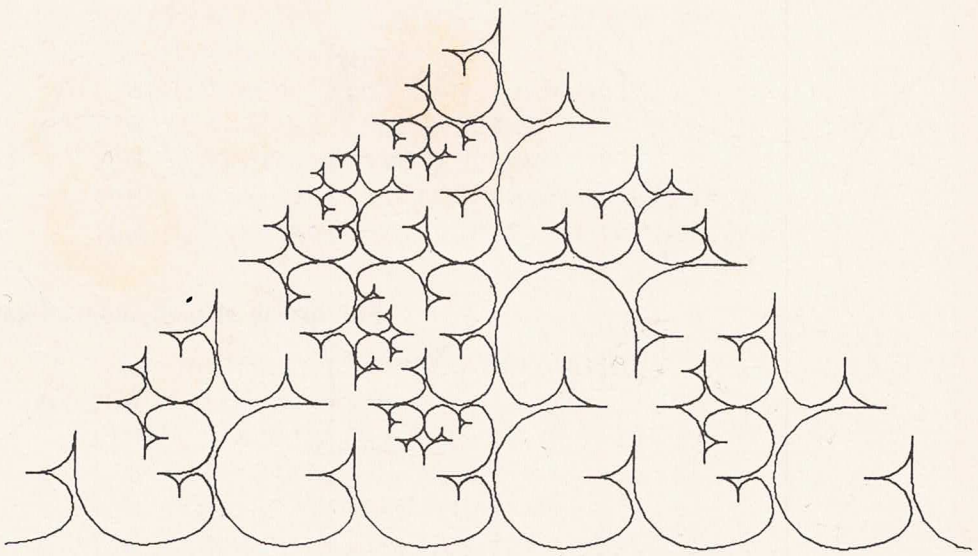
```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=4:K=3:S=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.5:L(1)=.45:L(2)=.45:L(3)=.5
150 A(0)=0:A(1)=PI/2:A(2)=-PI/2:A(3)=0
```



136

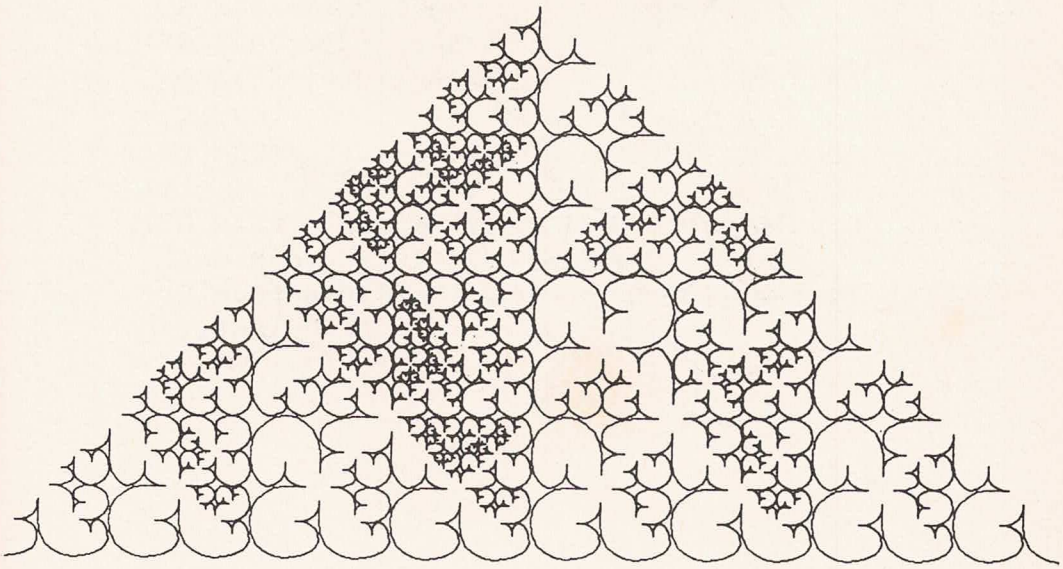
137

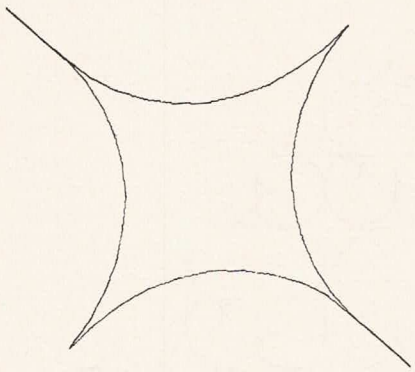




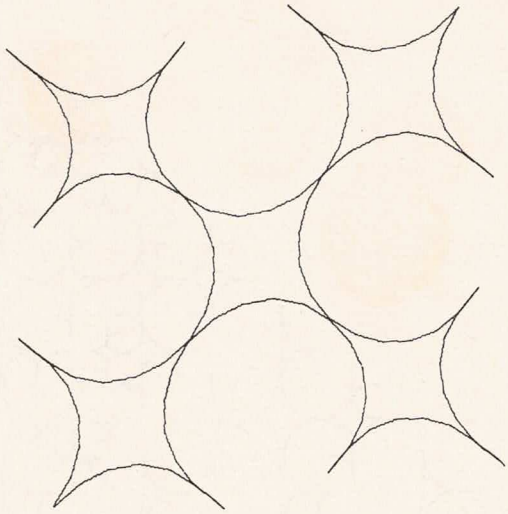
138

139



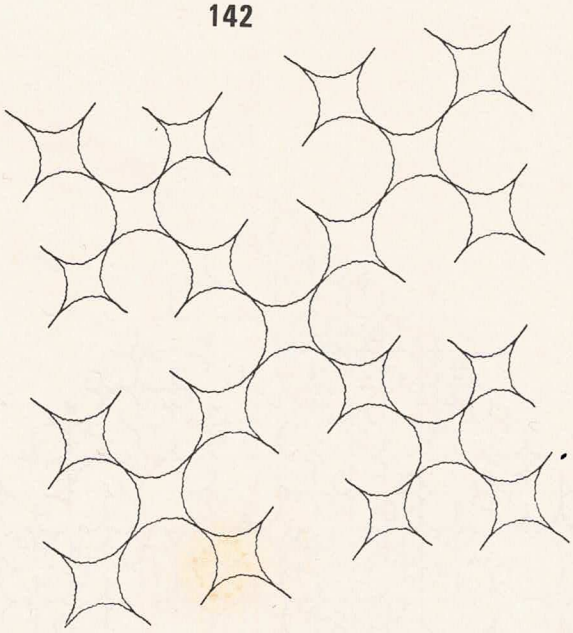


140



141

142



Modifications pour le dessin 143

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=4:K=4:S=3
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.5:L(1)=.45:L(2)=.45:L(3)=.5
150 A(0)=0:A(1)=PI/2:A(2)=-PI/2:A(3)=0
```

Modifications pour le dessin 144

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=4:K=5:S=3
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.5:L(1)=.45:L(2)=.45:L(3)=.5
150 A(0)=0:A(1)=PI/2:A(2)=-PI/2:A(3)=0
```

Modifications pour le dessin 145

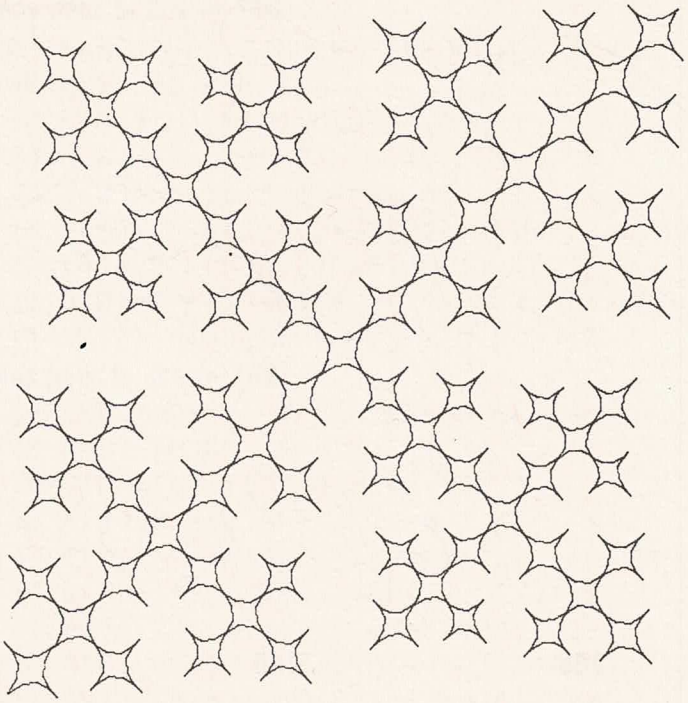
```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=13:K=1:S=8
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP
130 Y(0)=NP:Y(1)=0
140 L(0)=.4:L(1)=.4:L(2)=.2:L(3)=.2:L(4)
=.2:L(5)=.2:L(6)=.4:L(7)=.4
142 L(8)=.2:L(9)=.2:L(10)=.2:L(11)=.2:L(
12)=.2
150 A(0)=0:A(1)=PI/2:A(2)=0:A(3)=-PI/2:A
(4)=0:A(5)=-PI/2:A(6)=-PI:A(7)=-PI/2
152 A(8)=0:A(9)=PI/2:A(10)=0:A(11)=PI/2:
A(12)=0
```

Modifications pour le dessin 146

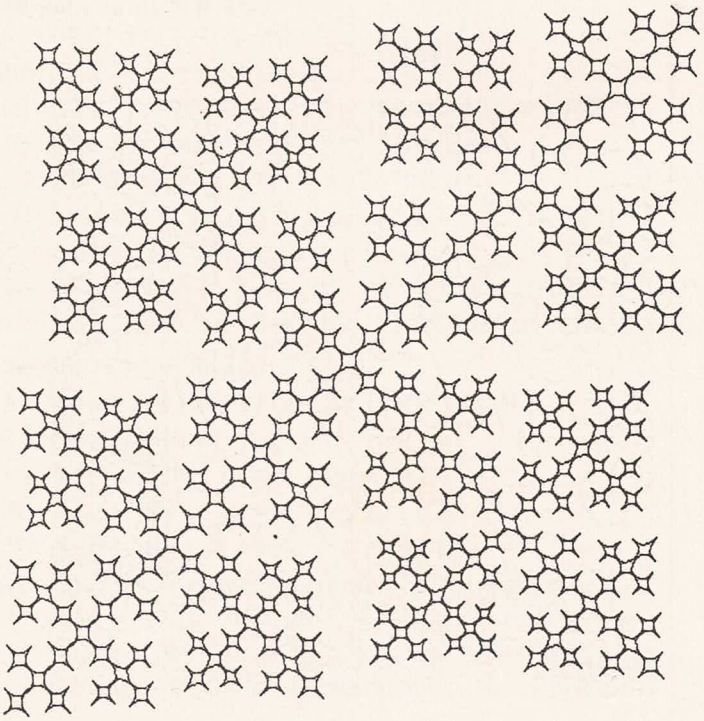
```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=13:K=2:S=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP
130 Y(0)=NP:Y(1)=0
140 L(0)=.4:L(1)=.4:L(2)=.2:L(3)=.2:L(4)
=.2:L(5)=.2:L(6)=.4:L(7)=.4
142 L(8)=.2:L(9)=.2:L(10)=.2:L(11)=.2:L(
12)=.2
150 A(0)=0:A(1)=PI/2:A(2)=0:A(3)=-PI/2:A
(4)=0:A(5)=-PI/2:A(6)=-PI:A(7)=-PI/2
152 A(8)=0:A(9)=PI/2:A(10)=0:A(11)=PI/2:
A(12)=0
```

Modifications pour le dessin 147

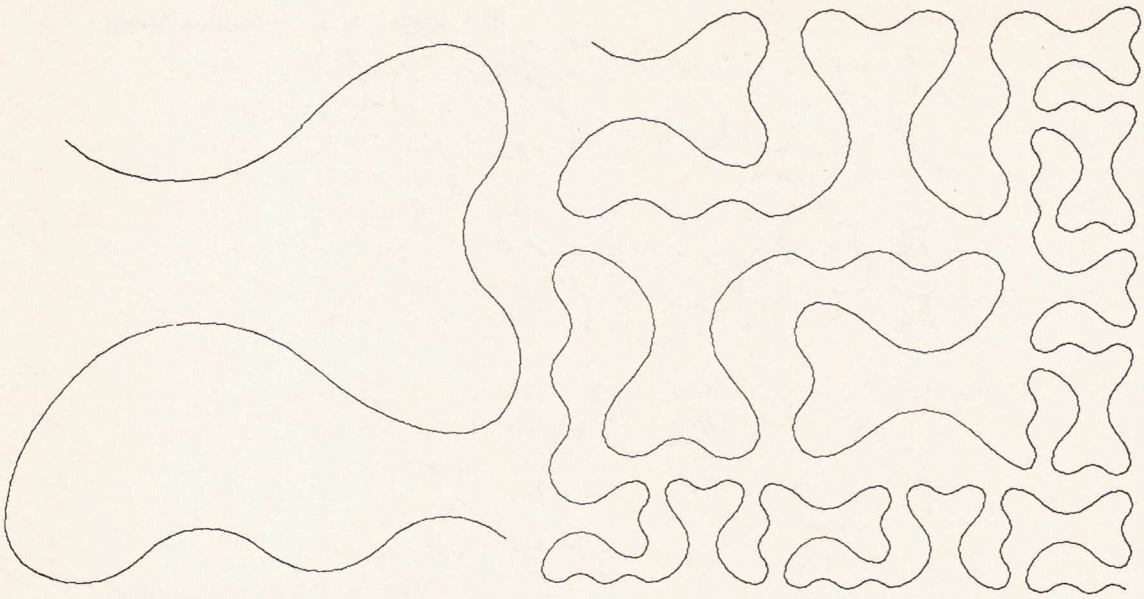
```
10 'FRACTALES SIMPLES ARRONDIES
100 M=1:N=13:K=3:S=3
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP
130 Y(0)=NP:Y(1)=0
140 L(0)=.4:L(1)=.4:L(2)=.2:L(3)=.2:L(4)
=.2:L(5)=.2:L(6)=.4:L(7)=.4
142 L(8)=.2:L(9)=.2:L(10)=.2:L(11)=.2:L(
12)=.2
150 A(0)=0:A(1)=PI/2:A(2)=0:A(3)=-PI/2:A
(4)=0:A(5)=-PI/2:A(6)=-PI:A(7)=-PI/2
152 A(8)=0:A(9)=PI/2:A(10)=0:A(11)=PI/2:
A(12)=0
```



143

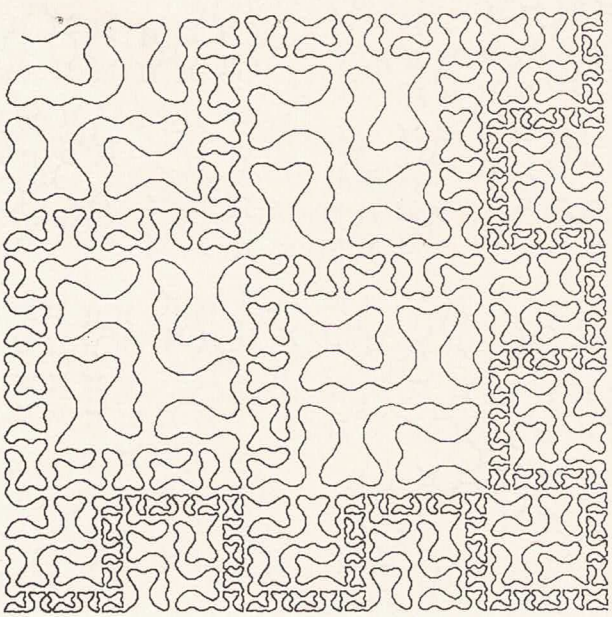


144



145 146

147



Modifications pour le dessin 148

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=8:K=1:S=8
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.4:L(1)=.2*SQR(2):L(2)=.2:L(3)=
.2:L(4)=.2:L(5)=.2:L(6)=L(1):L(7)=.2
150 A(0)=0:A(1)=PI/4:A(2)=PI/2:A(3)=PI:A
(4)=-PI/2:A(5)=0:A(6)=-PI/4:A(7)=0
```

Modifications pour le dessin 149

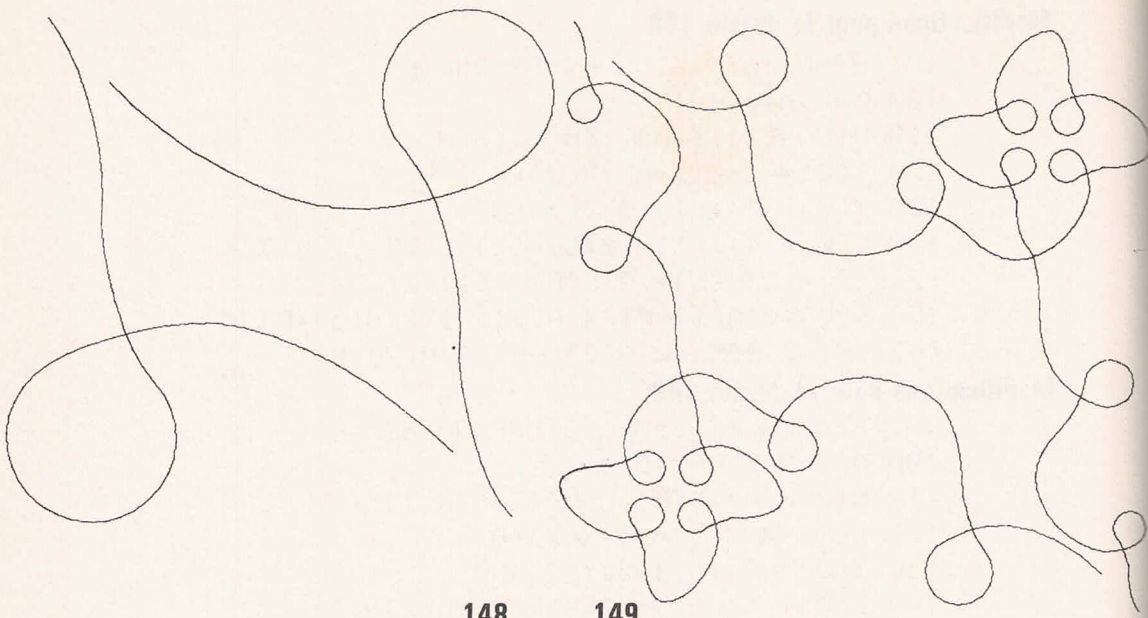
```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=8:K=2:S=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.4:L(1)=.2*SQR(2):L(2)=.2:L(3)=
.2:L(4)=.2:L(5)=.2:L(6)=L(1):L(7)=.2
150 A(0)=0:A(1)=PI/4:A(2)=PI/2:A(3)=PI:A
(4)=-PI/2:A(5)=0:A(6)=-PI/4:A(7)=0
```

Modifications pour le dessin 150

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=8:K=3:S=3
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.4:L(1)=.2*SQR(2):L(2)=.2:L(3)=
.2:L(4)=.2:L(5)=.2:L(6)=L(1):L(7)=.2
150 A(0)=0:A(1)=PI/4:A(2)=PI/2:A(3)=PI:A
(4)=-PI/2:A(5)=0:A(6)=-PI/4:A(7)=0
```

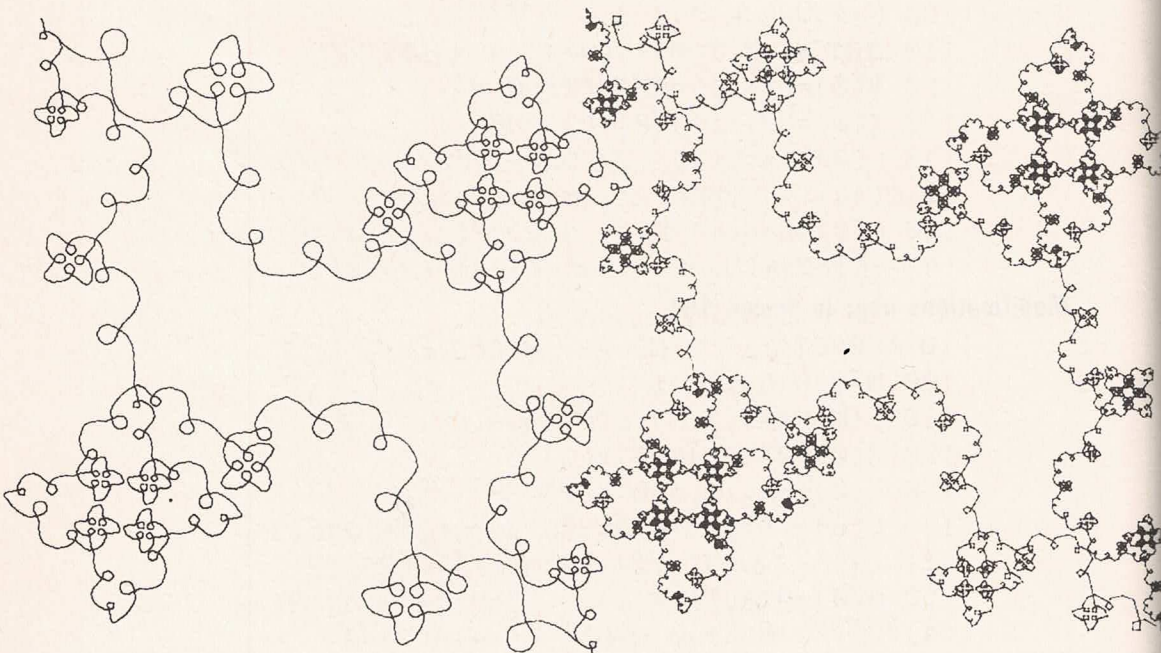
Modifications pour le dessin 151

```
10 'FRACTALES SIMPLES ARRONDIES
100 M=2:N=8:K=4:S=1
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 X(0)=0:X(1)=NP:X(2)=0
130 Y(0)=NP:Y(1)=0:Y(2)=NP
140 L(0)=.4:L(1)=.2*SQR(2):L(2)=.2:L(3)=
.2:L(4)=.2:L(5)=.2:L(6)=L(1):L(7)=.2
150 A(0)=0:A(1)=PI/4:A(2)=PI/2:A(3)=PI:A
(4)=-PI/2:A(5)=0:A(6)=-PI/4:A(7)=0
```



148 149

150 151



150

Le programme FRACTALES SIMPLES DÉFORMÉES

Les courbes fractales constituent des objets de base intéressants qu'il peut être amusant de « triturer » un peu. C'est ce que permet de faire ce programme basé sur le programme FRACTALES SIMPLES.

Les lignes modifiées sont les lignes 120, 130, 260, 485. Un sous-programme (lignes 5 000 - 5 500) est introduit. Son principe est le suivant :

à partir des coordonnées XØ, YØ du point A de la fenêtre [0, NP] x [0, NP] on calcule les coordonnées du point correspondant H dans le carré [— 1, 1] x [— 1, 1] qui est plus facilement manipulable (ligne 5 010).

Ensuite aux lignes 5 020, 5 030, on calcule la distance DH de ce point au centre 0 du carré [— 1, + 1] x [— 1, + 1], et l'angle AH du vecteur OH avec l'horizontale (DH et AH sont ce qu'on appelle les coordonnées polaires de H).

A la ligne 5 100, on effectue une transformation sur DH et AH, puis on calcule les coordonnées du point transformé dans la fenêtre [0, NP] x [0, NP].

Le dessin 152 correspond à un étirement vers le centre du flocon de Von Koch : DH = DH * DH.

Le dessin 153 correspond à une rotation de plus en plus forte quand on s'éloigne du centre, puis à un étirement vers le centre :
AH = AH + PI * DH : DH = DH * DH * DH * DH.

Les autres dessins utilisent des transformations sur les coordonnées polaires que l'on trouve toujours localisées sur la ligne 5 100. A partir du dessin 160, la déformation n'est plus faite sur un flocon de Von Koch mais sur une fractale du type des dessins 132 et 133.

Les dessins que permet de faire ce programme sont infiniment nombreux, avec un peu d'expérience, vous pourrez créer d'extraordinaires graphismes jamais vus.

Programme de base « FRACTALES SIMPLES DÉFORMÉES »

```
10 'FRACTALES SIMPLES DEFORMEES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 M=3:N=4:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 FOR IJ=0 TO 3:X(IJ)=NP/2*(1+SIN(2*IJ
*PI/3))
130 Y(IJ)=NP/2*(1+COS(2*IJ*PI/3)):NEXT I
J
```

```
140 L(0)=1/3:L(1)=L(0):L(2)=L(0):L(3)=L(
0)
150 A(0)=0:A(1)=PI/3:A(2)=-A(1):A(3)=0
200 FOR II=0 TO M-1
210 XD=X(II):YD=Y(II):XA=X(II+1):YA=Y(II
+1)
250 X0=XD:Y0=YD
260 GOSUB 5000
270 LPRINT "M";X1;",";Y1
280 IFXA<>XD THEN A0=ATN((YA-YD)/(XA-XD)
) ELSE A0=PI/2*SGN(YA-YD)
285 IF (XA-XD)<0 THEN A0=A0+PI
290 L0=SQR((XA-XD)^2+(YA-YD)^2)
300 FOR I=0 TO N^K-1
310 LL=L0:AA=A0:T1=I
390 IF K=0 THEN 470
400 FOR J=K-1 TO 0 STEP -1
410 R=N^J:T2=INT(T1/R)
430 AA=AA+A(T2):LL=LL*L(T2)
440 T1=T1-T2*R
450 NEXT J
470 X0=X0+LL*COS(AA)
480 Y0=Y0+LL*SIN(AA)
485 GOSUB 5000
490 LPRINT"D";X1;",";Y1
500 NEXT I
600 NEXT II
1000 END
5000 'SOUS-PROGRAMME DE DEFORMATION
5010 XH=X0/NP*2-1:YH=Y0/NP*2-1
5020 DH=SQR(XH*XH+YH*YH)
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2 ELSE AH=PI/2*SGN(YH)
5100 DH=DH*DH
5200 X1=INT((DH*COS(AH)+1)*NP/2)
5210 Y1=INT((DH*SIN(AH)+1)*NP/2)
5500 RETURN
```

Modifications pour le dessin 152

```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=4
```

Modifications pour le dessin 153

```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=5
5100 AH=AH+PI*DH:DH=DH*DH*DH*DH
```

Modifications pour le dessin 154

```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=4
5100 DH=DH^5:AH=AH+PI/4*SIN(2*PI*DH)
```

Modifications pour le dessin 155

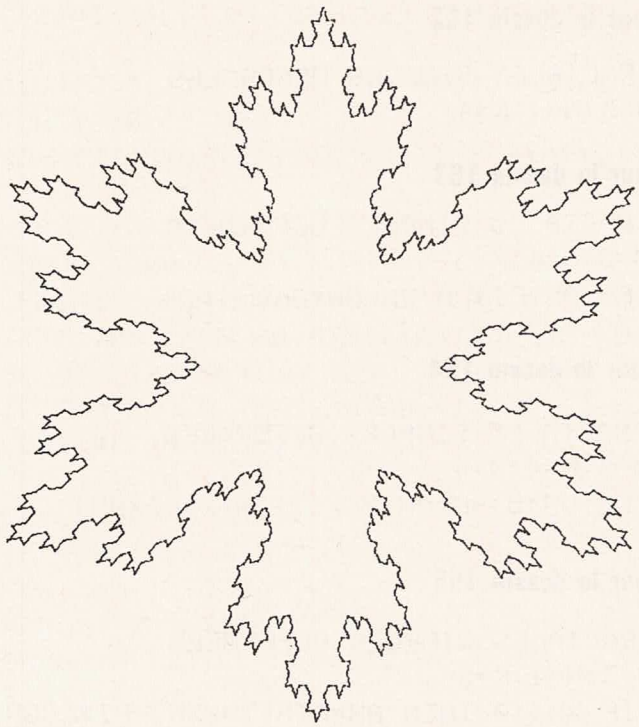
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=5
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH) ELSE AH=PI/2*SGN(YH)
5100 DH=DH^6:AH=AH*AH*AH/PI/PI
```

Modifications pour le dessin 156

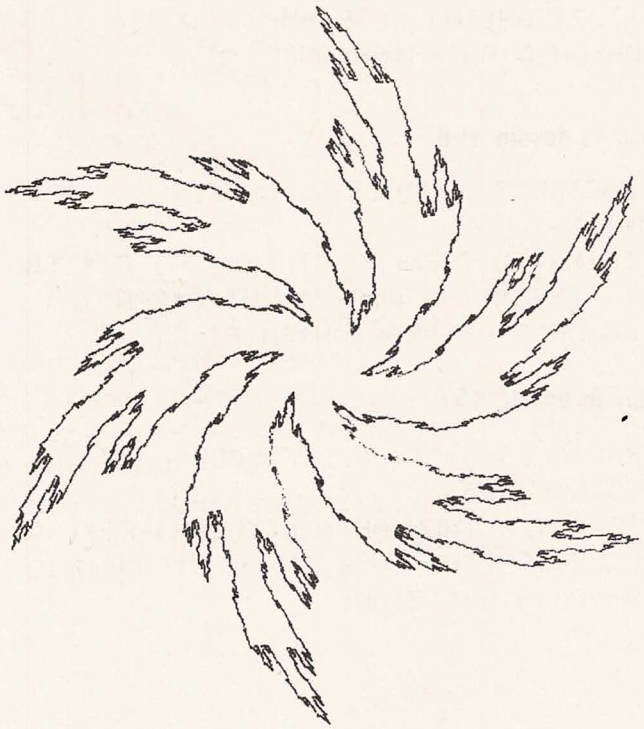
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=5
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH) ELSE AH=PI/2*SGN(YH)
5100 DH=DH^6:AH=4*AH*AH*AH/PI/PI
```

Modifications pour le dessin 157

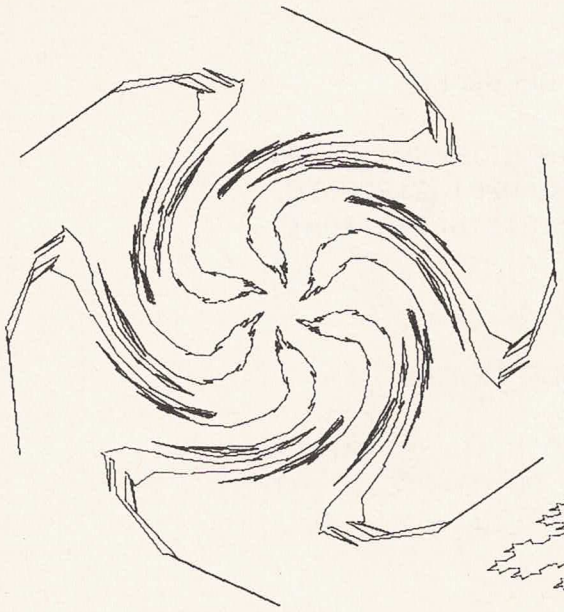
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=5
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH) ELSE AH=PI/2*SGN(YH)
5100 DH=DH^5:AH=10*AH
```



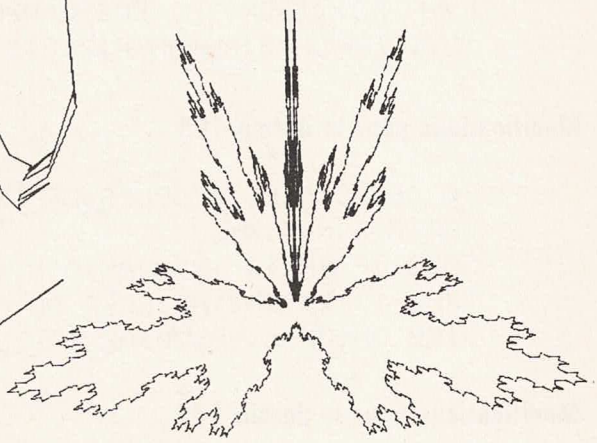
152



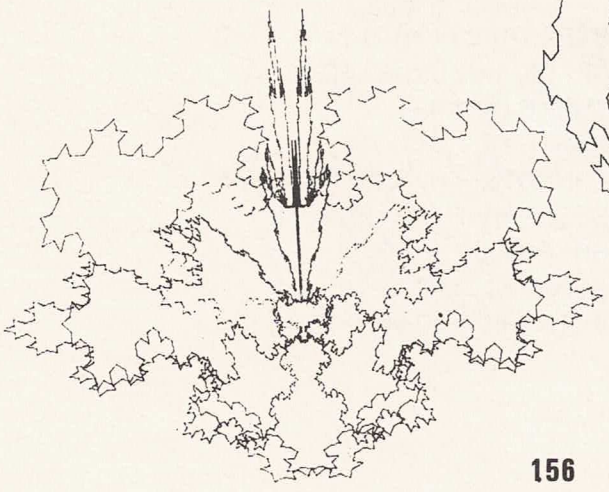
153



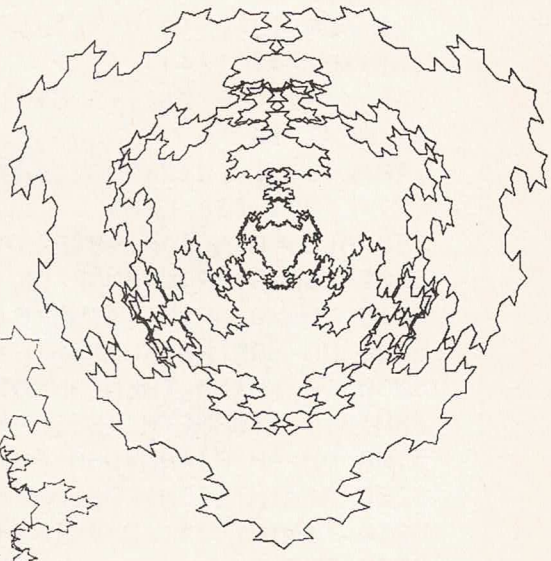
154



155



156



157

Modifications pour le dessin 158

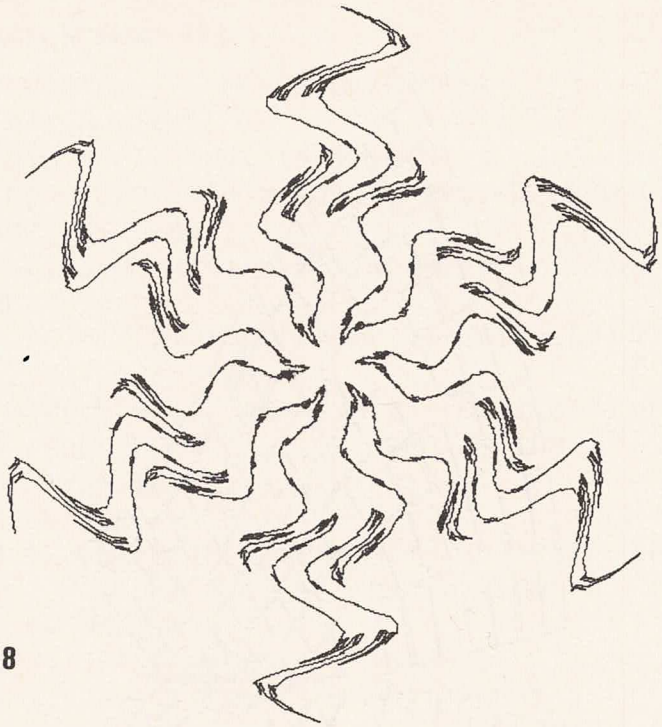
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=5
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH) ELSE AH=PI/2*SGN(YH)
5100 DH=DH^5:AH=AH+PI/18*SIN(6*PI*DH)
```

Modifications pour le dessin 159

```
10 'FRACTALES SIMPLES DEFORMEES
100 M=3:N=4:K=5
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH):ELSE AH=PI/2*SGN(YH)
5100 DH=DH^5:AH=20*AH
```

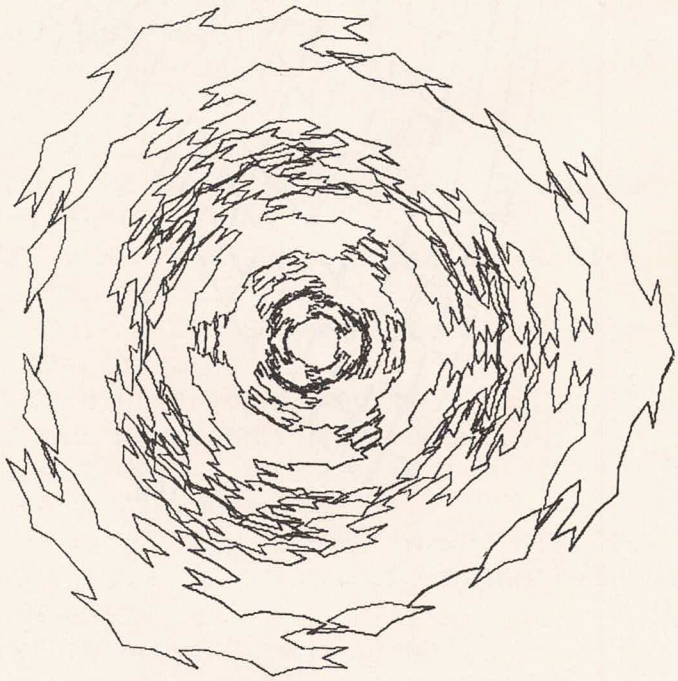
Modifications pour le dessin 160

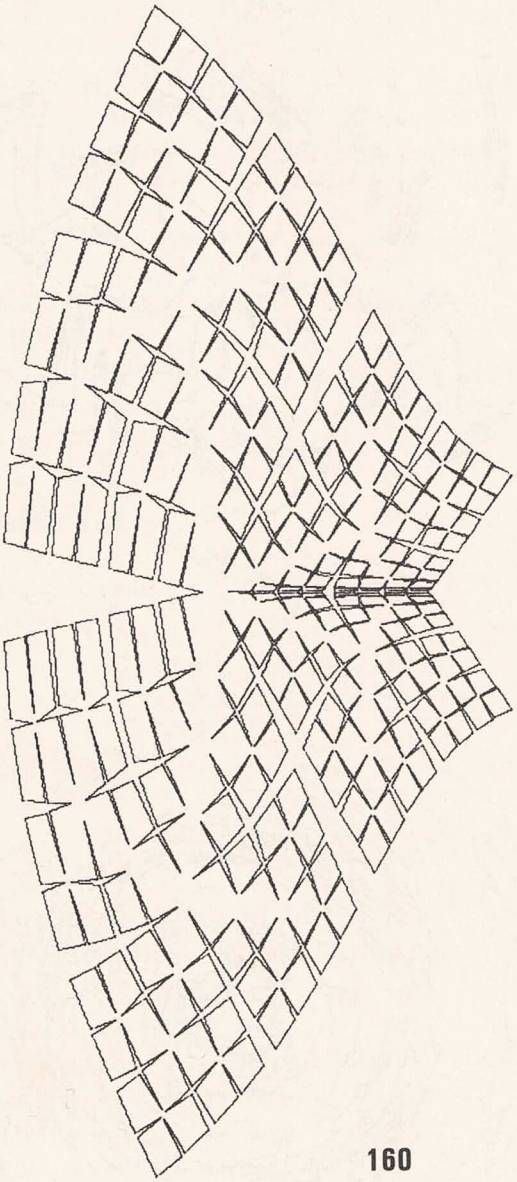
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=4:N=4:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 FOR IJ=0 TO 4:X(IJ)=NP/2*(1+5/6*COS(
2*IJ*PI/4+PI/4))
130 Y(IJ)=NP/2*(1+5/6*SIN(2*IJ*PI/4+PI/4
)):NEXT IJ
140 L(0)=1/(2+2*COS(.48*PI)):L(1)=L(0):L
(2)=L(0):L(3)=L(0)
150 A(0)=0:A(1)=.48*PI:A(2)=-A(1):A(3)=0
5000 'SOUS-PROGRAMME DE DEFORMATION
5010 XH=X0/NP*2-1:YH=Y0/NP*2-1
5020 DH=SQR(XH*XH+YH*YH)
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH):ELSE AH=PI/2*SGN(YH)
5100 AH=AH/PI*AH*SGN(AH)
5200 X1=INT((DH*COS(AH)+1)*NP/2)
5210 Y1=INT(2*((DH*SIN(AH)+1)*NP/2))
5500 RETURN
```



158

159





160

Modifications pour le dessin 161

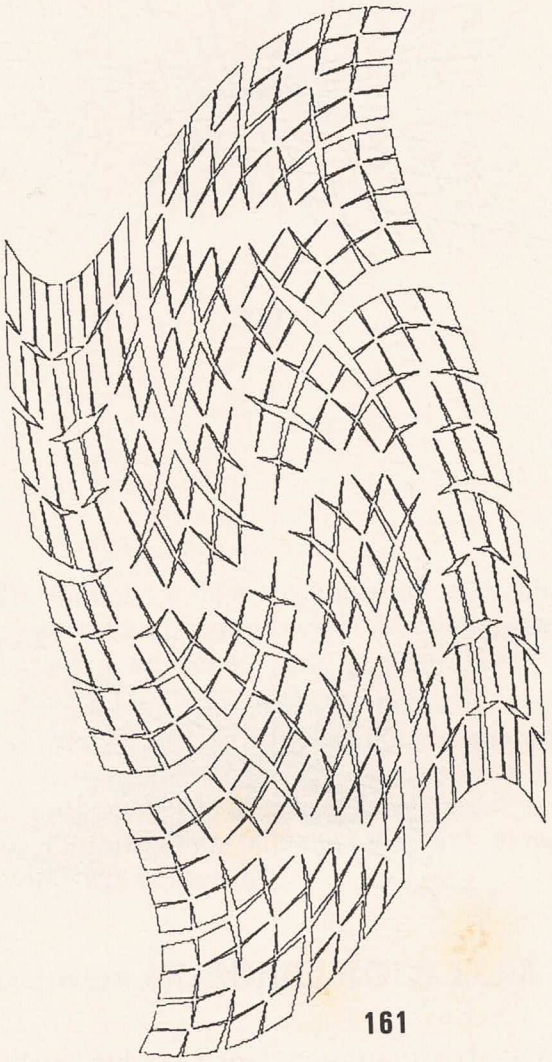
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=4:N=4:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 FOR IJ=0 TO 4:X(IJ)=NP/2*(1+5/6*COS(
2*IJ*PI/4+PI/4))
130 Y(IJ)=NP/2*(1+5/6*SIN(2*IJ*PI/4+PI/4
)):NEXT IJ
140 L(0)=1/(2+2*COS(.48*PI)):L(1)=L(0):L
(2)=L(0):L(3)=L(0)
150 A(0)=0:A(1)=.48*PI:A(2)=-A(1):A(3)=0
5000 'SOUS-PROGRAMME DE DEFORMATION
5010 XH=X0/NP*2-1:YH=Y0/NP*2-1
5020 DH=SQR(XH*XH+YH*YH)
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH):ELSE AH=PI/2*SGN(YH)
5100 AH=AH+PI/2*(1-DH)
5200 X1=INT((DH*COS(AH)+1)*NP/2)
5210 Y1=INT(2*((DH*SIN(AH)+1)*NP/2))
5500 RETURN
```

Modifications pour le dessin 162

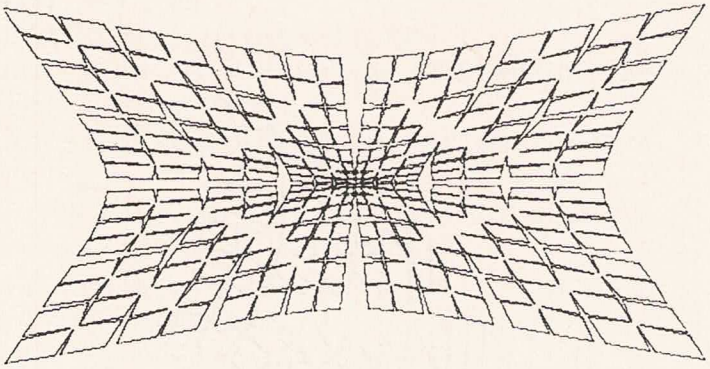
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=4:N=4:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 FOR IJ=0 TO 4:X(IJ)=NP/2*(1+5/6*COS(
2*IJ*PI/4+PI/4))
130 Y(IJ)=NP/2*(1+5/6*SIN(2*IJ*PI/4+PI/4
)):NEXT IJ
140 L(0)=1/(2+2*COS(.48*PI)):L(1)=L(0):L
(2)=L(0):L(3)=L(0)
150 A(0)=0:A(1)=.48*PI:A(2)=-A(1):A(3)=0
5000 'SOUS-PROGRAMME DE DEFORMATION
5010 XH=X0/NP*2-1:YH=Y0/NP*2-1
5020 DH=SQR(XH*XH+YH*YH)
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH):ELSE AH=PI/2*SGN(YH)
5100 DH=DH*DH
5200 X1=INT((DH*COS(AH)+1)*NP/2)
5210 Y1=INT(2*((DH*SIN(AH)+1)*NP/2))
5500 RETURN
```

Modifications pour le dessin 163

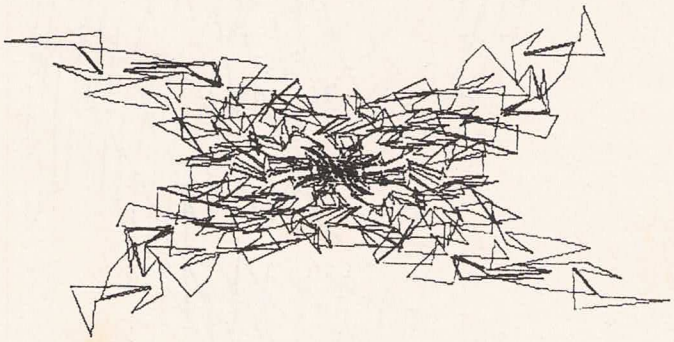
```
10 'FRACTALES SIMPLES DEFORMEES
100 M=4:N=4:K=4
110 DIM X(M),Y(M),L(N-1),A(N-1)
120 FOR IJ=0 TO 4:X(IJ)=NP/2*(1+5/6*COS(
2*IJ*PI/4+PI/4))
130 Y(IJ)=NP/2*(1+5/6*SIN(2*IJ*PI/4+PI/4
)):NEXT IJ
140 L(0)=1/(2+2*COS(.48*PI)):L(1)=L(0):L
(2)=L(0):L(3)=L(0)
150 A(0)=0:A(1)=.48*PI:A(2)=-A(1):A(3)=0
5000 'SOUS-PROGRAMME DE DEFORMATION
5010 XH=X0/NP*2-1:YH=Y0/NP*2-1
5020 DH=SQR(XH*XH+YH*YH)
5030 IF XH<>0 THEN AH=ATN(YH/XH)-PI*(SGN
(XH)-1)/2*SGN(YH):ELSE AH=PI/2*SGN(YH)
5100 DH=DH^3:AH=AH+PI/18*SIN(6*PI/DH)
5200 X1=INT((DH*COS(AH)+1)*NP/2)
5210 Y1=INT(2*((DH*SIN(AH)+1)*NP/2))
5500 RETURN
```



161



162



163

8 Quadrillages élastiques

Le principe des déformations a déjà été utilisé dans les chapitres 2 et 7. Ici nous allons le reconsidérer un peu plus systématiquement en prenant comme objet de base un simple quadrillage.

Les résultats obtenus rappellent, couleurs en moins, les tableaux de Vasarely (voir les références [8] [17] [33] [39]), on pourra quand même noter que l'informatique nous permet des essais très variés qu'à la main, il est peu concevable d'envisager. Nous voyons sur cet exemple qu'une fois quelques difficultés franchies, l'informatique libère la créativité, et autorise des recherches esthétiques impossibles sans elle.

Le programme QUADRILLAGES ÉLASTIQUES

Imaginons qu'on ait une grille en caoutchouc mou parfaitement extensible : on pourra éloigner les points du centre (dessin 164), ou, au contraire les rapprocher du centre (dessin 165) ou les faire pivoter (dessin 166) ou faire ces opérations simultanément et avec des amplitudes variées (dessins 166-173). A la place d'une grille, on peut partir d'un simple réseau de lignes parallèles (dessins 174-176).

Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Début de la boucle principale. Quand L = 1 les lignes horizontales se dessinent. Quand L = 0 ce sont les lignes verticales.

Ligne 110

Début de la boucle secondaire. A chaque parcours de cette boucle, une ligne se dessine.

Ligne 120

Début de la boucle tertiaire. A chaque parcours de cette boucle, un segment d'une ligne se dessine.

Ligne 200

Calcul des coordonnées du J-ième point de la ligne numéro I, dans le carré $[-1, +1] \times [-1, +1]$.

Ligne 210

On permute X et Y lorsque L = 1, ce qui donne une ligne horizontale au lieu d'une ligne verticale.

Lignes 300 à 320

Calcul des coordonnées polaires du J-ième point de la ligne numéro I. Voir le programme FRACTALES SIMPLES DÉFORMÉES.

Ligne 400

Opération sur les coordonnées polaires. C'est ce qui crée la déformation. En supprimant cette ligne on obtiendrait une simple grille.

On remarquera qu'on ne déforme la grille que pour les points dont la distance au centre du carré est plus petite que 1. Le contraste entre la zone modifiée du quadrillage et la zone laissée intacte accroît l'effet de relief.

Lignes 500, 510

Calcul des coordonnées du point transformé, cette fois-ci dans le carré [0, NP] x [0, NP].

Ligne 600

Tracé du J-ième segment de la ligne numéro I lorsque J>0. Sinon positionnement de la plume.

Lignes 700, 1 000

Fin des 3 boucles imbriquées. Fin du programme.

Programme de base « QUADRILLAGES ÉLASTIQUES »

```
10 'QUADRILLAGES ELASTIQUES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 FOR L=0 TO 1
110 FOR I=0 TO 20
120 FOR J=0 TO 20
200 X=I/10-1:Y=J/10-1
210 IF L=1 THEN T=X:X=Y:Y=T
300 DI=SQR(X*X+Y*Y)
310 IF X<>0 THEN AN=ATN(Y/X) ELSE AN=PI/
2*SGN(Y)
320 IF X<0 THEN AN=AN+PI
400 IF DI<1 THEN DI=DI^.3
500 X=DI*COS(AN):Y=DI*SIN(AN)
510 X%=INT(NP/2*(1+.95*X)):Y%=INT(NP/2*(
1+.95*Y))
600 IF J=0 THEN LPRINT"M";X%;",";Y%
610 IF J>0 THEN LPRINT"D";X%;",";Y%
700 NEXT J:NEXT I:NEXT L
1000 END
```

Modifications pour le dessin 164

```
10 'QUADRILLAGES ELASTIQUES
```

Modifications pour le dessin 165

```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN DI=DI^3
```

Modifications pour le dessin 166

```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+PI/2*(1-DI)
```

Modifications pour le dessin 167

```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+3.5*(1-DI):DI=DI^
 3
```

Modifications pour le dessin 168

```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+2*PI*(1-DI)
```

Modifications pour le dessin 169

```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+PI*(1-DI):DI=DI^3
```

Modifications pour le dessin 170

```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+PI/2*SIN(PI*(1-DI
)):DI=DI^.2
```

Modifications pour le dessin 171

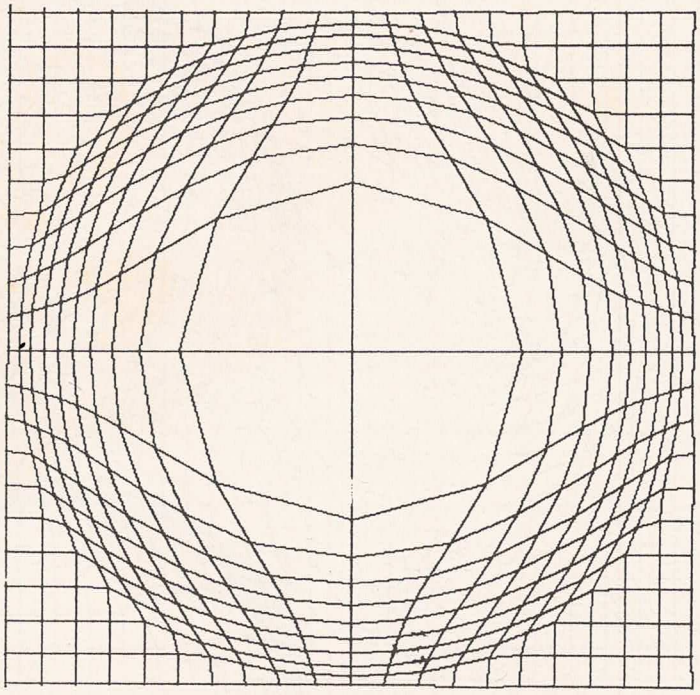
```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+PI/4*SIN(2*PI*(1-
DI))
```

Modifications pour le dessin 172

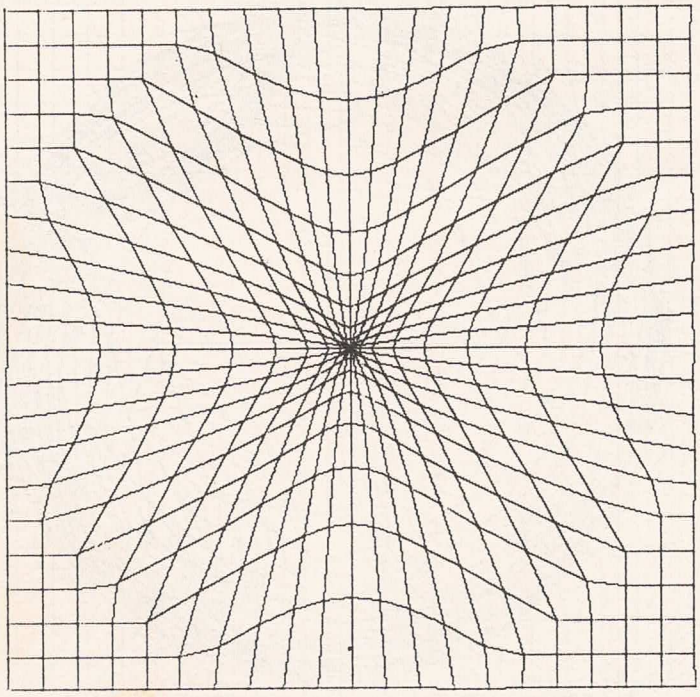
```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+PI/2*SIN(PI*(1-DI
)):DI=DI^2
```

Modifications pour le dessin 173

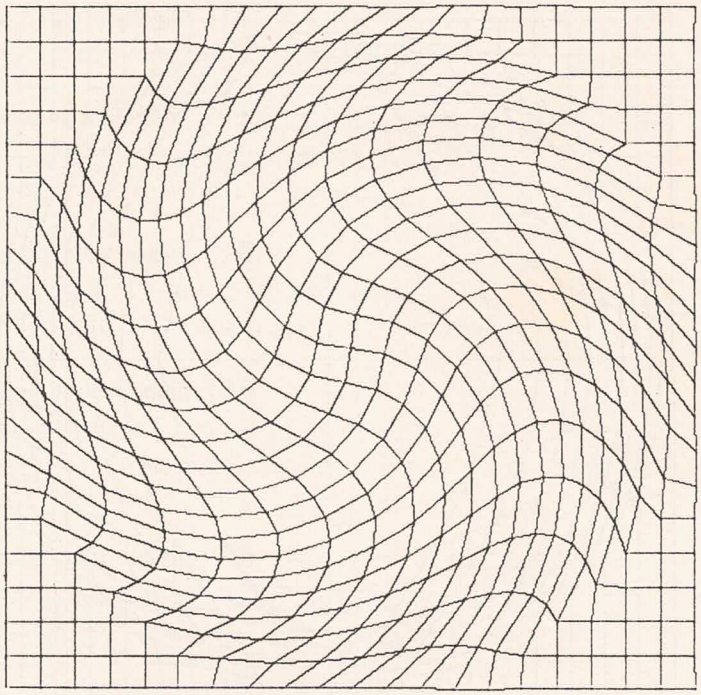
```
10 'QUADRILLAGES ELASTIQUES
400 IF DI<1 THEN AN=AN+PI/2*SIN(PI*(1-DI
))
```



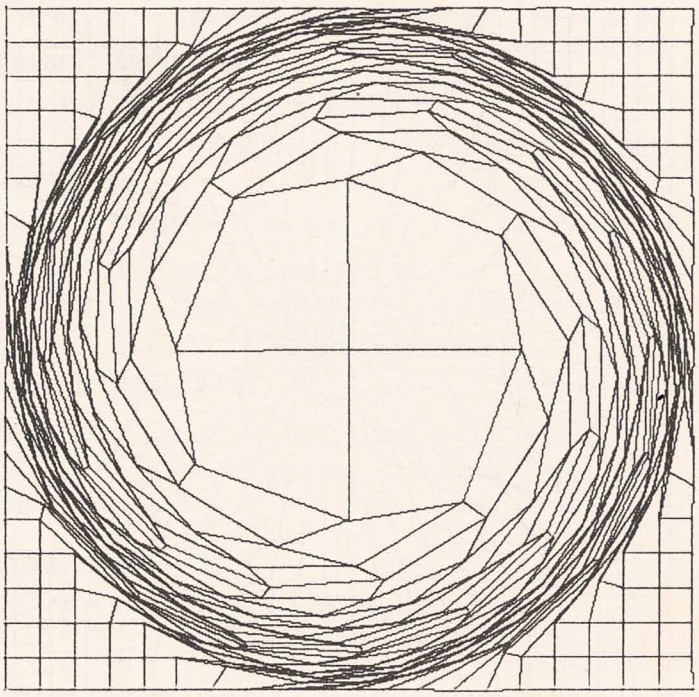
164



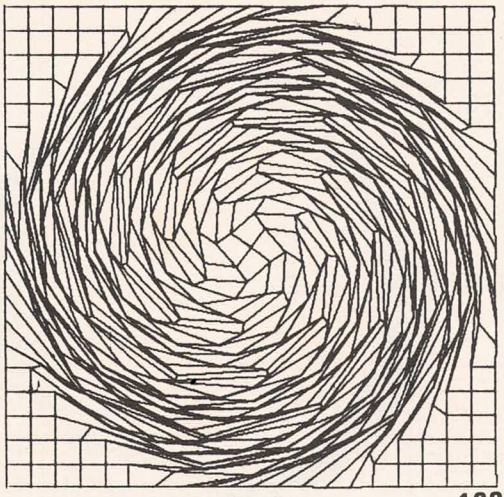
165



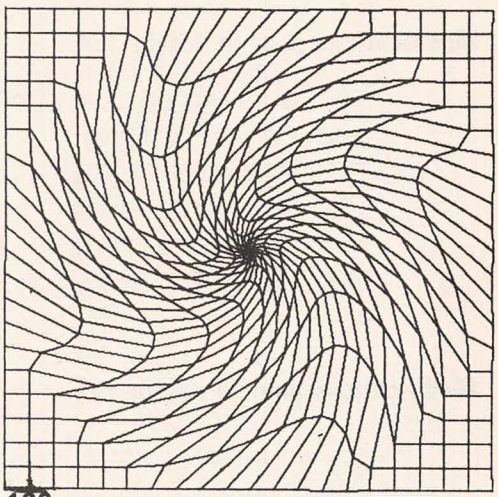
166



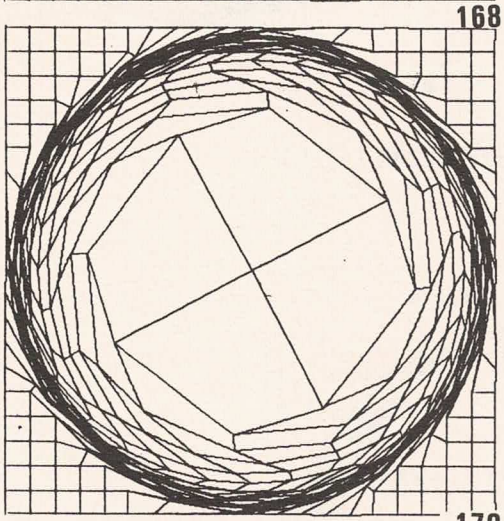
167



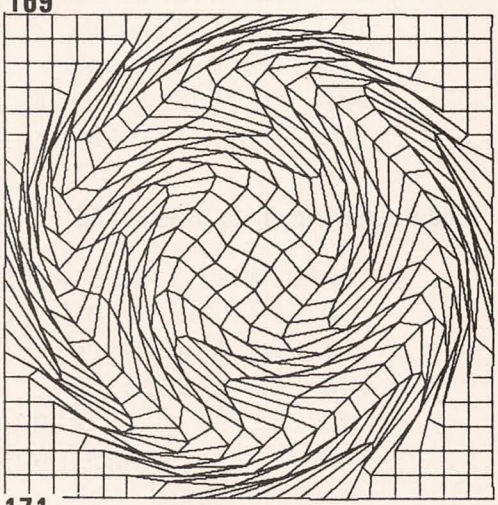
168



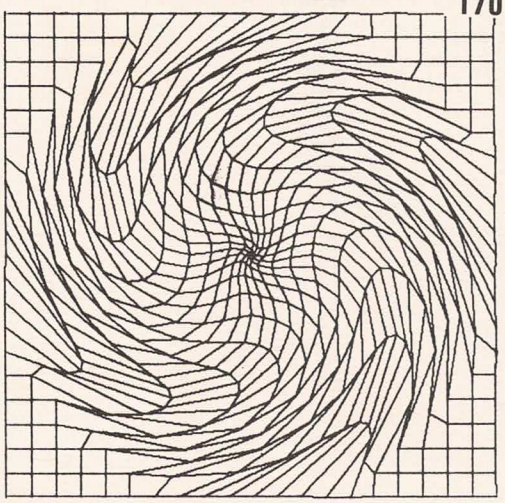
169



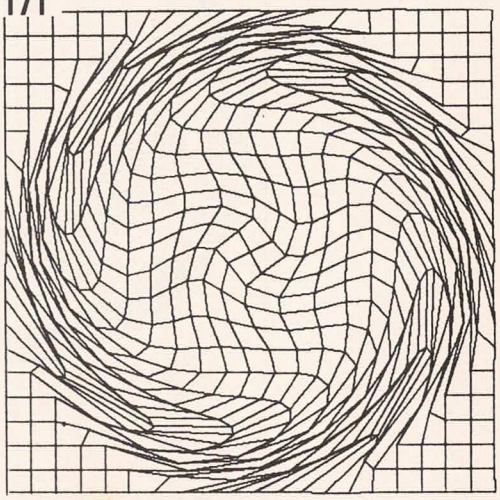
170



171



172



173

Modifications pour le dessin 174

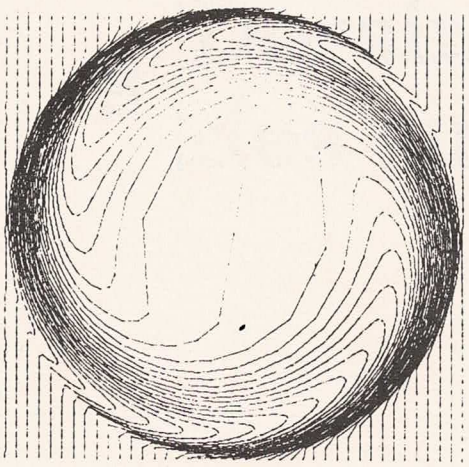
```
10 'QUADRILLAGES ELASTIQUES
100 FOR L=0 TO 0
110 FOR I=0 TO 40
120 FOR J=0 TO 40
200 X=I/20-1:Y=J/20-1
400 IF DI<1 THEN AN=AN+PI*(1-DI):DI=DI^.
3
```

Modifications pour le dessin 175

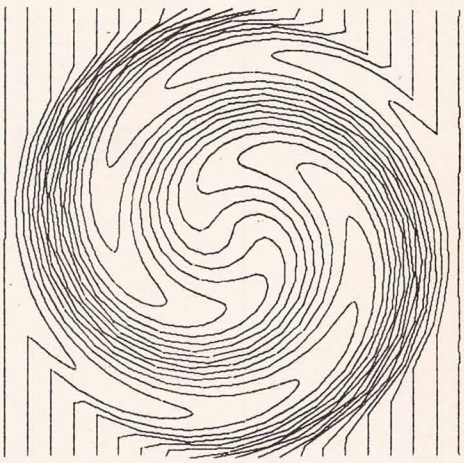
```
10 'QUADRILLAGES ELASTIQUES
100 FOR L=0 TO 0
110 FOR I=0 TO 20
120 FOR J=0 TO 40
200 X=I/10-1:Y=J/20-1
400 IF DI<1 THEN AN=AN+PI*2*(1-DI)
```

Modifications pour le dessin 176

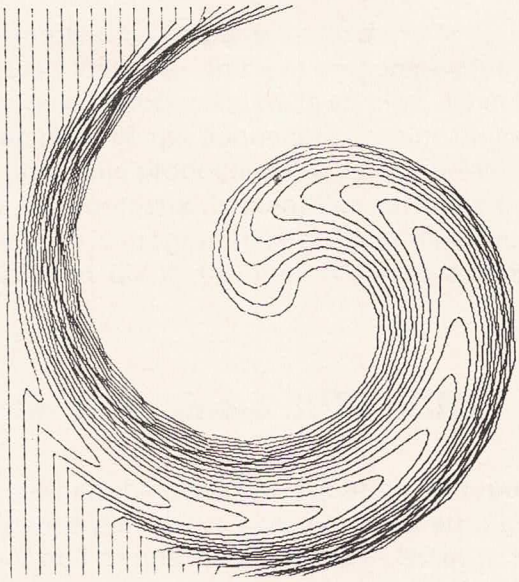
```
10 'QUADRILLAGES ELASTIQUES
100 FOR L=0 TO 0
110 FOR I=0 TO 20
120 FOR J=0 TO 40
200 X=I/20-1:Y=J/20-1
400 IF DI<1 THEN AN=AN+PI*2*(1-DI)
```



174



175



176

9 Dessiner des surfaces

Encore plus que les courbes, les surfaces mathématiques présentent des qualités esthétiques intéressantes et variées.

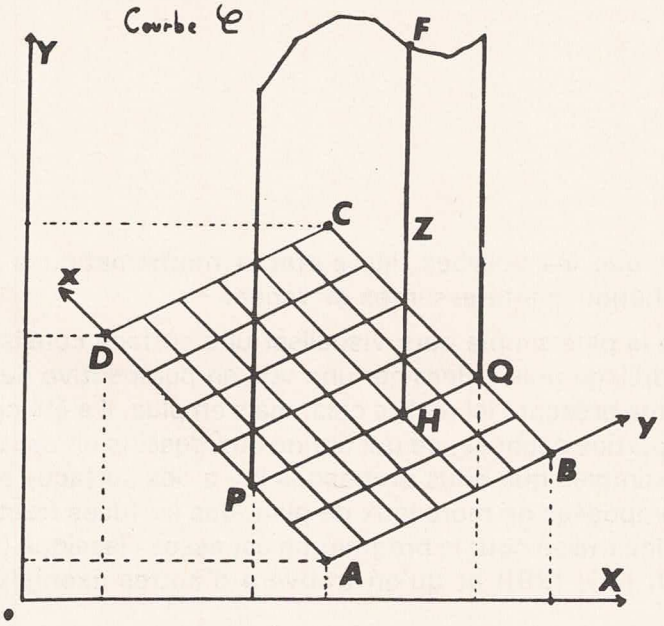
La méthode la plus simple pour visualiser une surface consiste à tracer dessus un quadrillage puis à dessiner une vue en perspective de ce quadrillage. Le programme présenté ici réalise cela, mais en plus, il a été conçu de façon à effacer les parties cachées, ce qui donne aux dessins un aspect très réaliste. Parmi les exemples que nous proposons il y a des surfaces algébriques, des surfaces composées de morceaux de plan, des surfaces fractales. Indiquons que le principe utilisé pour le programme est assez classique (voir les références [9] [10] [20] [29]) et qu'on trouvera d'autres exemples dans la référence [29].

Le programme SURFACES

Toutes les surfaces définissables par une équation du type Z = f (X, Y) vont pouvoir être représentées par ce programme. On s'arrange pour que X et Y soient compris entre 0 et 1 et que Z varie entre — NP et + NP (quelle que soit la surface envisagée, on peut se ramener à cette situation), et on place l'équation (ou les équations) définissant Z = f (X, Y) dans le sous-programme entre les lignes 5 000 et 5 500.

L'algorithme destiné à effacer les parties cachées est basé sur le principe suivant :

on imagine que le papier sur lequel on dessine est découpé en bandes verticales de même largeur ; il y a M % = NP/PA bandes verticales (dans l'adaptation de ce programme, il faut faire attention à ce que M % soit bien un entier). Deux vecteurs (déclarés à la ligne 200) MA % et MI % sont destinés à contenir pour chaque bande verticale la valeur maximale et la valeur minimale de la zone où il ne faut plus dessiner. Les deux vecteurs sont initialisés à la ligne 100 puis, au cours du tracé, (qui s'effectue des courbes les plus proches aux courbes les plus éloignées) ces vecteurs sont tenus à jour ce qui permet de savoir pour chaque segment du quadrillage de la surface s'il faut ou non le dessiner (car bien sûr, s'il est caché il ne peut l'être que par un morceau de surface dessiné avant lui).



Ligne 50

Initialisation — Voir le chapitre 0.

Ligne 100

Fixation des paramètres du tracé :

N % : nombre de lignes du quadrillage.

PA : finesse du pas de l'algorithme de dessin.
Lorsque le pas est petit, le tracé est plus précis mais plus long. Arrangez-vous toujours pour que ce pas soit un diviseur de NP.

E1 : si E1 = 1, une seule série de lignes parallèles sera dessinée ; si E1 = 2, deux séries de lignes parallèles seront dessinées, ce qui donnera un quadrillage de la surface.

E2 : Si E2 = 0, les parties cachées de la surface ne seront pas dessinées ; si E2 = 1, les parties cachées seront dessinées.

Ligne 110

Fixation des coordonnées des points A, B, C, D (voir la figure d'explication) qui déterminent la position de vue. Toutes ces coordonnées sont à prendre entre 0 et NP.

Ligne 200

Calcul du nombre de bandes verticales servant à l'algorithme d'effacement des parties cachées. Déclaration des tableaux MA % et MI % destinés à contenir pour chaque bande verticale la valeur maximale et la valeur minimale de la zone de la bande où il ne faut plus dessiner.

Ligne 300

Initialisation de MA % et MI %.

Ligne 500

Début de la boucle principale. A chaque parcours de cette boucle, une courbe C au-dessus des points P et Q (voir figure) est dessinée.

Lignes 510 - 540

Calcul des coordonnées des points P et Q.

Ligne 550

Calcul de la valeur de X correspondant à l'abscisse des points du segment PQ. Lorsque E1 = 2 les lignes 300 - 1 050 sont parcourues deux fois (une fois pour chaque série de lignes parallèles du quadrillage). La première fois E3 = 0, la seconde E3 = 1 (voir la ligne 1 010).

Ligne 560

Calcul (dans le repère correspondant au papier) des abscisses des points P et Q en prenant comme unité de longueur la largeur d'une bande verticale.

Ligne 600

Début de la boucle secondaire. A chaque parcours de cette boucle, un segment de la courbe C est dessiné, ou parcouru pointe levée s'il est caché.

Ligne 610

Calcul de Y (ou de X quand E3 = 1) correspondant à l'ordonnée du point H. Les valeurs de X et de Y sont comprises entre 0 et 1.

Ligne 620

Appel du sous-programme qui à partir de X et Y calcule Z.

Lignes 630 - 640

Calcul des coordonnées de F (dans le repère correspondant au papier).

Lignes 700 - 760

Tracé d'un segment jusqu'au point F, ou simple déplacement, selon que F est dans une zone interdite ou pas. Mise à jour de MI % et MA %

Lignes 800 - 900

Fin des boucles.

Lignes 1 000 - 1 050

Fin du programme lorsque E1 = 1, ou renvoi en 300 pour le tracé du deuxième réseau de lignes parallèles (après permutation des coordonnées de D et B) lorsque E1 = 2.

Lignes 5 000 - 5 500

Sous-programme qui, à partir de X et Y, compris entre 0 et 1, calcule F (X, Y), compris entre — NP et + NP (ou compris dans un autre intervalle selon les contraintes de fenêtres, et selon le choix des points A, B, C, D).

Les dessins 177, 178, 179, 180, 181 illustrent les effets des paramètres E1 et E2 qui contrôlent le nombre de réseaux de lignes parallèles et l'effacement des parties cachées.

Les dessins 182, 183, 184, 185 sont obtenus avec des équations utilisant les fonctions trigonométriques. Les dessins 186, 187 utilisent un procédé (ligne 5 010) permettant la répétition d'une même forme.

Les dessins 188, 189 représentent des surfaces algébriques (voir la référence [29]).

Le dessin 190 est un grand classique (voir la référence [10]).

Les dessins 191, 192, 193, 194, 195 montrent des surfaces composées de morceaux de plans, on les obtient avec des « max » et des « min » de fonctions affines. L'équation correspondant au dessin 193 est par exemple : Z = max (min [1.5 — 6 I x — .5I, 1.5 — 6 I Y — .5 I], .5 — 2 I x — .5 I, .25 — IY — .5I).

Les dessins 196, 197 représentent des reliefs fractals.

Les dessins 198, 199, 200 représentent d'autres surfaces fractales.

REMARQUE.

Lorsqu'on prend PA petit (ce qui est souhaitable pour obtenir des tracés précis), l'exécution des dessins peut devenir très longue (plusieurs heures). Ce n'est pas vraiment étonnant car tout algorithme de traitement des parties cachées en trois dimensions est nécessairement coûteux en calculs (et aussi en mémoires).

Programme de base « SURFACES »

```
10 'SURFACES
50 LPRINT CHR$(18):LPRINT"M0,-600":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 N%=6:PA=NP/16:E1=2:E2=1
110 XA=NP/2:YA=NP/16:XB=NP*7/8:YB=NP/4:X
C=NP/2:YC=NP*5/8:XD=NP/8:YD=NP*7/16
200 M%=NP/PA:DIM MA%(M%),MI%(M%)
300 FOR I=0 TO M%:MA%(I)=-5*NP:MI%(I)=5*
NP:NEXT I
500 FOR I=0 TO N%
510 XP=(I*XD+(N%-I)*XA)/N%
520 YP=(I*YD+(N%-I)*YA)/N%
530 XQ=(I*XC+(N%-I)*XB)/N%
540 YQ=(I*YC+(N%-I)*YB)/N%
550 IF E3=1 THEN Y=I/N% ELSE X=I/N%
560 I1=INT(XP/PA):I2=INT(XQ/PA):G=SGN(I2
-I1)
600 FOR J=I1 TO I2 STEP G
610 IF E3=1 THEN X=(J-I1)/(I2-I1) ELSE Y
=(J-I1)/(I2-I1)
620 GOSUB 5000
630 XF=INT(J*PA)
640 YF=((J-I1)*YQ+(I2-J)*YP)/(I2-I1)+Z:Y
F=INT(YF)
700 IF J=I1 THEN LPRINT"M";XF;",";YF
710 IF E2=1 THEN 760
720 IF YF>MI%(J) AND YF<MA%(J) THEN LPRI
NT"M";XF;",";YF:GOTO 800
```

```
740 IF YF>MA%(J) THEN MA%(J)=YF
750 IF YF<MI%(J) THEN MI%(J)=YF
760 LPRINT"D";XF;",";YF
800 NEXT J
900 NEXT I
1000 IF E1=1 THEN END
1010 E3=1:E1=1
1020 UU=XD:XD=XB:XB=UU
1030 UU=YD:YD=YB:YB=UU
1050 GOTO 300
5000 ' SOUS-PROGRAMME POUR Z=F(X,Y)
5010 Z=NP/3*SIN(PI*Y)*SIN(PI*X)
5500 RETURN
```

Modifications pour le dessin 177

```
10 'SURFACES
```

Modifications pour le dessin 178

```
10 'SURFACES
100 N%=60:PA=NP/160:E1=1:E2=0
```

Modifications pour le dessin 179

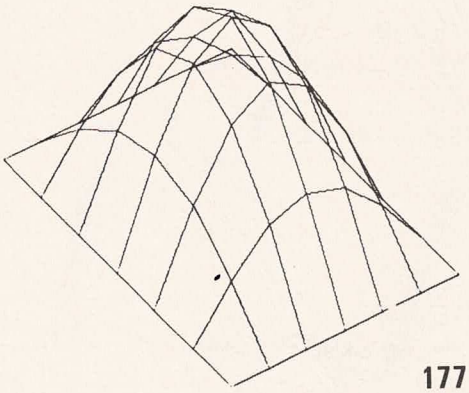
```
10 'SURFACES
100 N%=60:PA=NP/160:E1=1:E2=1
```

Modifications pour le dessin 180

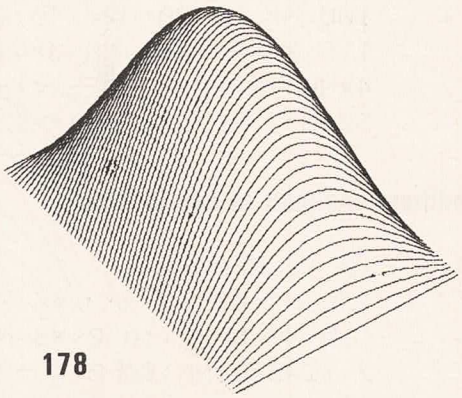
```
10 'SURFACES
100 N%=60:PA=NP/160:E1=2:E2=0
```

Modifications pour le dessin 181

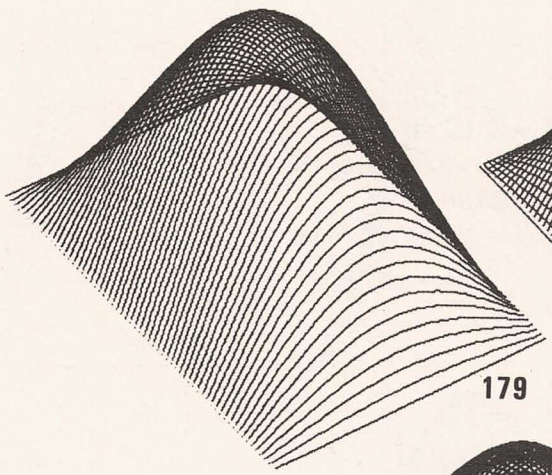
```
10 'SURFACES
100 N%=60:PA=NP/160:E1=2:E2=1
```



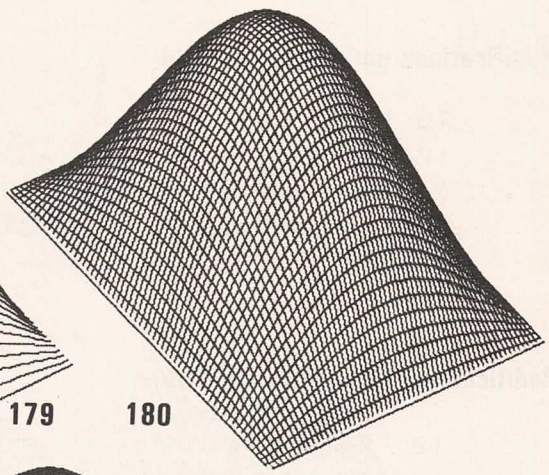
177



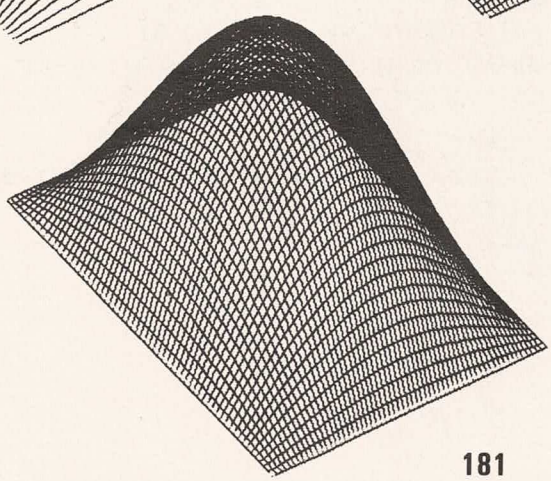
178



179



180



181

Modifications pour le dessin 182

```
10 'SURFACES
100 N%=60:PA=NP/240:E1=1:E2=0
110 XA=0:YA=0:XB=3/4*NP:YB=0:XC=NP:YC=5/
4*NP:XD=NP/4:YD=5/4*NP
5010 Z=NP/5*SIN(3*PI*Y)*SIN(4*PI*X)
```

Modifications pour le dessin 183

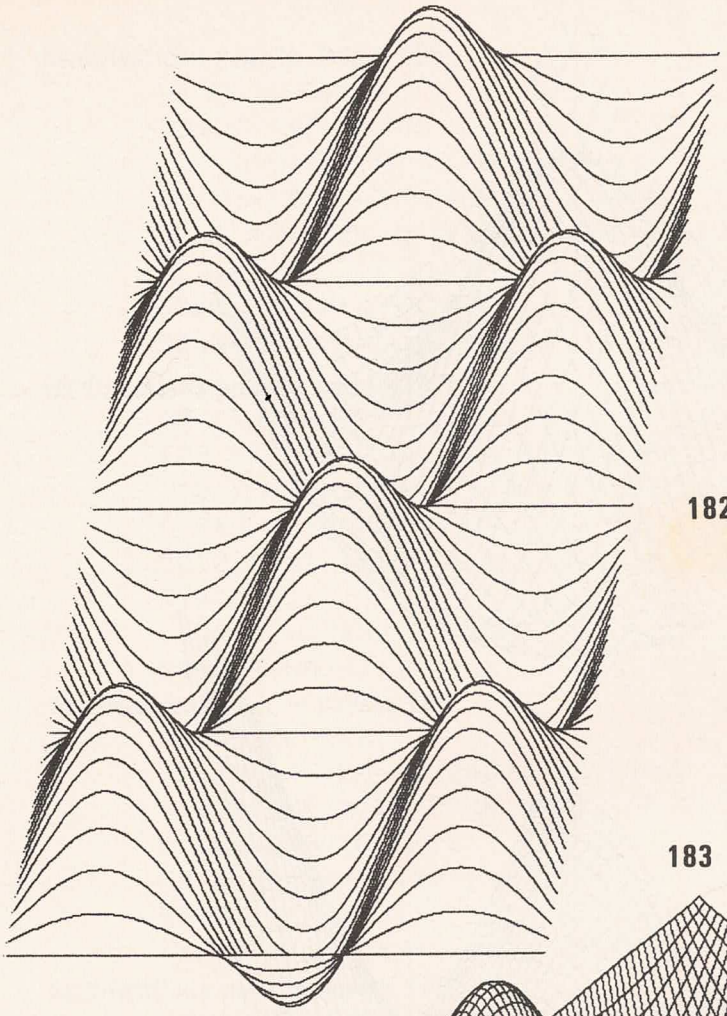
```
10 'SURFACES
100 N%=60:PA=NP/240:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=5/12*NP:XC=NP/
2:YC=5/6*NP:XD=0:YD=5/12*NP
5010 Z=NP/4*SIN(2*PI*Y)*SIN(3*PI*X)
```

Modifications pour le dessin 184

```
10 'SURFACES
100 N%=30:PA=NP/240:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=5/12*NP:XC=NP/
2:YC=5/6*NP:XD=0:YD=5/12*NP
5010 DI=7*SQR((X-.5)*(X-.5)+(Y-.5)*(Y-.5
)):Z=COS(DI)*NP/5
```

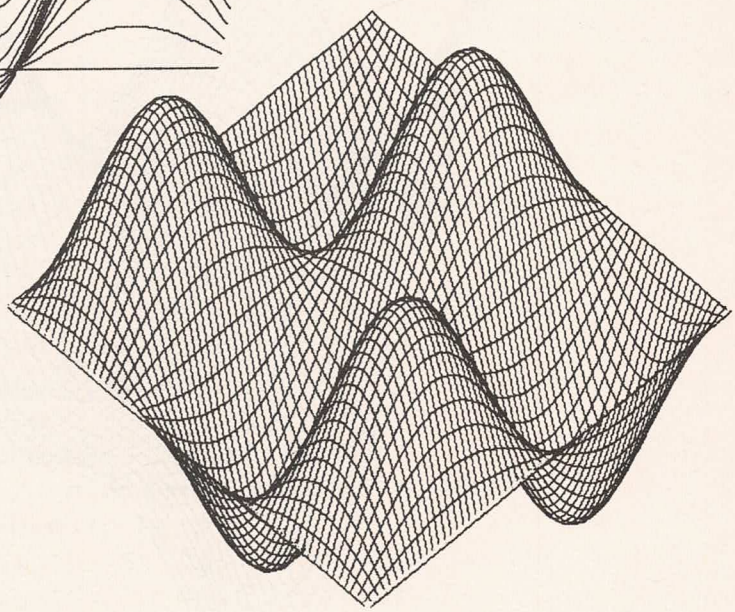
Modifications pour le dessin 185

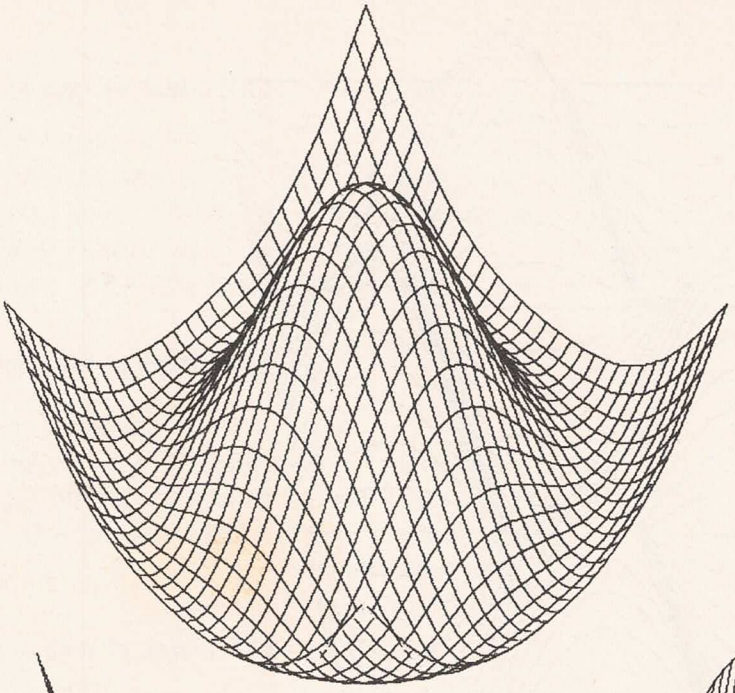
```
10 'SURFACES
100 N%=40:PA=NP/240:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/2:XC=NP/2:Y
C=NP:XD=0:YD=NP/2
5010 X7=2*X-1:Y7=2*Y-1
5020 IF X7*Y7<>0 THEN Z=3*NP/4*X7*Y7*(X7
*Y7-Y7*Y7)/(X7*X7+Y7*Y7) ELSE Z=0
```



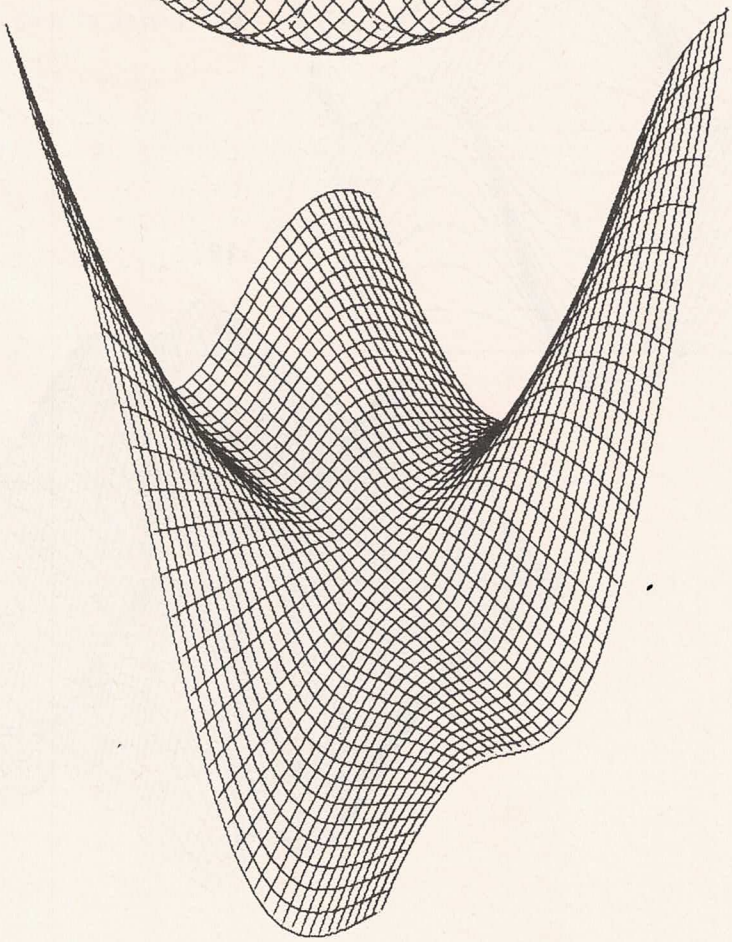
182

183





184



185

Modifications pour le dessin 186

```
10 'SURFACES
100 N%=30:PA=NP/160:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/3:XC=NP/2:Y
C=2*NP/3:XD=0:YD=NP/3
5010 X7=(3*X-INT(3*X))*2-1:Y7=(2*Y-INT(2
*Y))*2-1
5020 DI=X7*X7+Y7*Y7
5030 Z=NP/4*(1-DI^.2)
```

Modifications pour le dessin 187

```
10 'SURFACES
100 N%=50:PA=NP/240:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/3:XC=NP/2:Y
C=2*NP/3:XD=0:YD=NP/3
5010 X7=(5*X-INT(5*X))*2-1:Y7=(5*Y-INT(5
*Y))*2-1
5020 DI=X7*X7+Y7*Y7
5030 Z=NP/4*(1-DI)*(.6-ABS(Y-.5))
```

Modifications pour le dessin 188

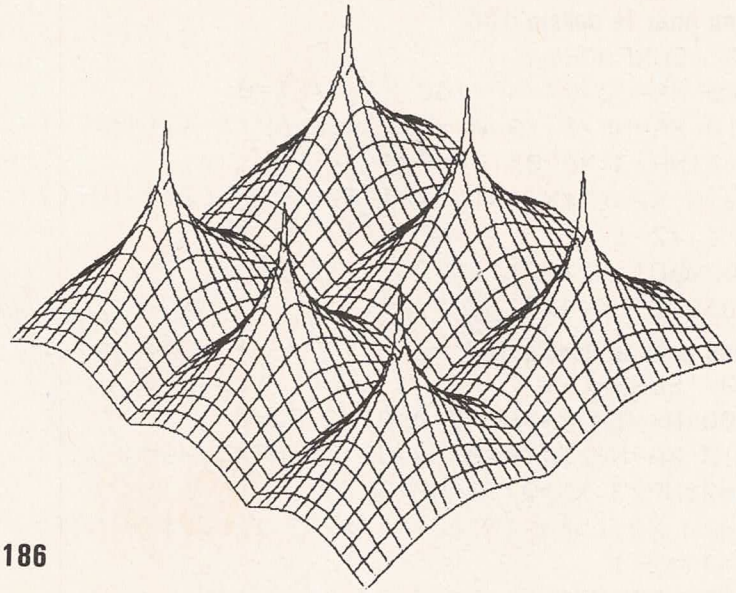
```
10 'SURFACES
100 N%=40:PA=NP/240:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/2:XC=NP/2:Y
C=NP:XD=0:YD=NP/2
5010 X7=2*X-1:Y7=2*Y-1
5020 IF X7*Y7<>0 THEN Z=3*NP/4*X7*Y7*(X7
*X7-Y7*Y7)/(X7*X7+Y7*Y7) ELSE Z=0
```

Modifications pour le dessin 189

```
10 'SURFACES
100 N%=80:PA=NP/480:E1=1:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/3:XC=NP/2:Y
C=2*NP/3:XD=0:YD=NP/3
5010 X7=3*Y-1.5:Y7=3*X-1.5
5020 IF X7*Y7<>0 THEN Z=NP/4*Y7*X7*X7/(Y
7*Y7+X7*X7*X7*X7) ELSE Z=0
```

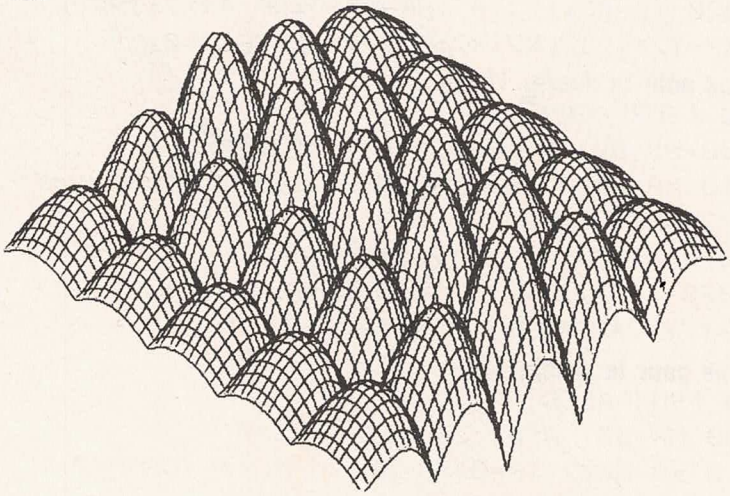
Modifications pour le dessin 190

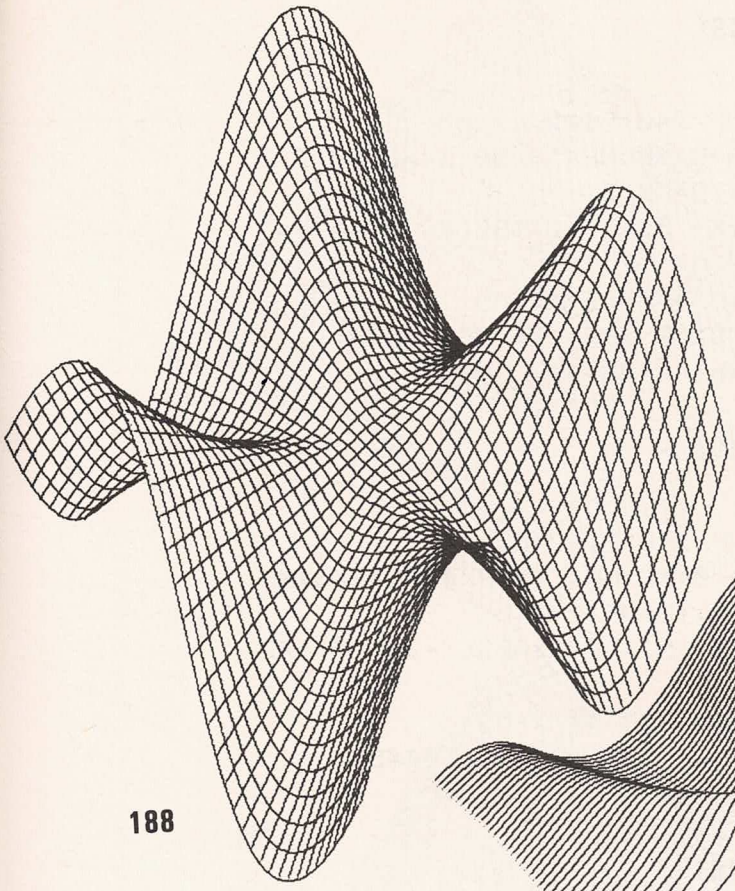
```
10 'SURFACES
100 N%=80:PA=NP/240:E1=1:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/3:XC=NP/2:Y
C=NP*2/3:XD=0:YD=NP/3
5010 DI=16*((X-.5)*(X-.5)+(Y-.5)*(Y-.5))
5020 Z=NP*5/12*COS(4*DI)*EXP(-DI)
```



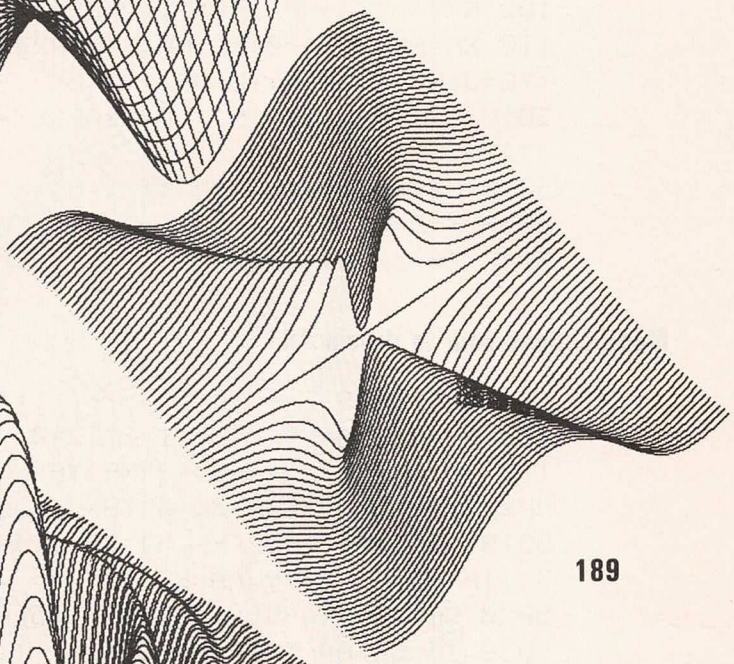
186

187

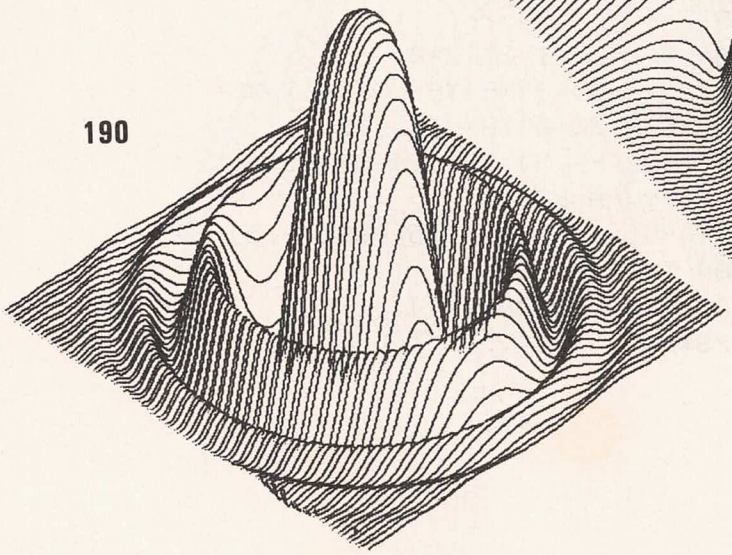




188



189



190

Modifications pour le dessin 191

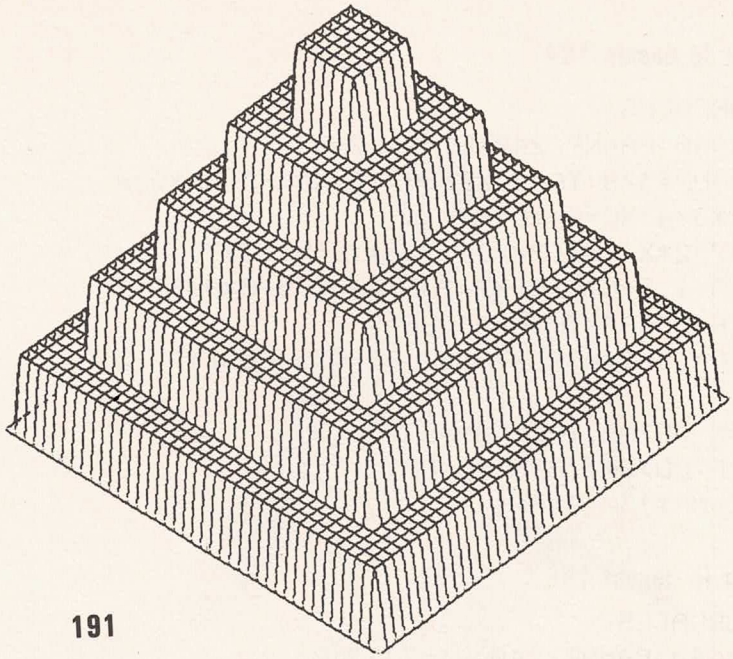
```
10 'SURFACES
100 N%=40:PA=NP/240:E1=2:E2=0
110 XA=NP/2:YA=0:XB=NP:YB=NP/3:XC=NP/2:Y
C=NP*2/3:XD=0:YD=NP/3
5010 X7=2*ABS(X-.5):Y7=2*ABS(Y-.5):M7=X7
:IF X7<Y7 THEN M7=Y7
5020 M8=INT(5*M7):M9=5*M7-M8
5030 Z=0 :IF M9>.8 THEN Z=(M9-.8)*5
5040 Z=-NP*5/48*(Z+M8)
```

Modifications pour le dessin 192

```
10 'SURFACES
100 N%=40:PA=NP/240:E1=2:E2=0
110 XA=NP/3:YA=0:XB=NP:YB=NP/4:XC=2/3*NP
:YC=3*NP/4:XD=0:YD=NP/2
5010 X7=2*ABS(X-.5):Y7=2*ABS(Y-.5):M7=X7
+Y7
5020 M8=INT(4*M7):M9=4*M7-M8
5030 Z=0:IF M9>.8 THEN Z=(M9-.8)*5
5040 Z=-5/48*NP*(Z+M8)
```

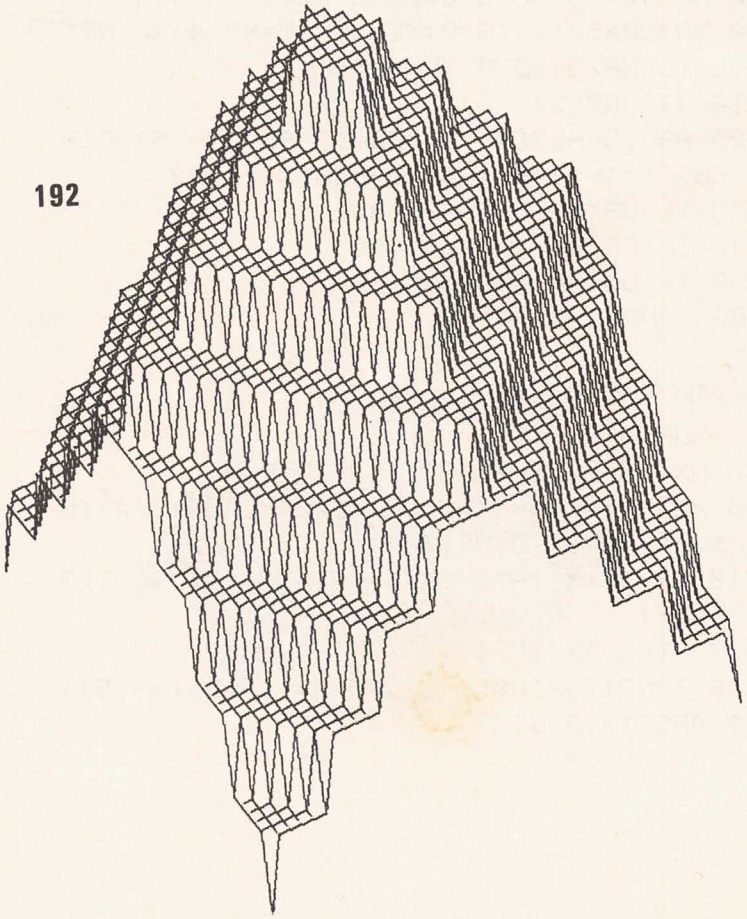
Modifications pour le dessin 193.

```
10 'SURFACES
100 N%=40:PA=NP/240:E1=2:E2=0
110 XA=NP*5/12:YA=0:XB=NP:YB=NP*5/12:XC=
NP*7/12:YC=NP*5/6:XD=0:YD=NP*5/12
5010 AA=1.5-6*ABS(X-.5):BB=1.5-6*ABS(Y-.
5):IF BB<AA THEN AA=BB
5020 CC=.5-2*ABS(X-.5):DD=.25-ABS(Y-.5)
5030 IF CC>AA THEN AA=CC
5040 IF DD>AA THEN AA=DD
5050 Z=NP*5/8*AA
```



191

192



Modifications pour le dessin 194

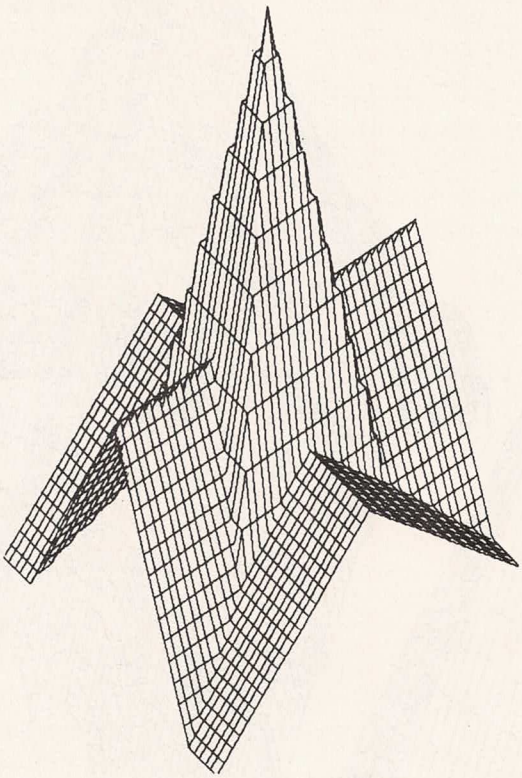
```
10 'SURFACES
100 N%=60:PA=NP/240:E1=2:E2=0
110 XA=NP*3/8:YA=0:XB=NP:YB=XA:XC=NP*5/8
:YC=NP*3/4:XD=0:YD=XA
5010 X7=2*X-INT(2*X):Y7=3*Y-INT(3*Y)
5020 AA=1.5-6*ABS(X7-.5):BB=1.5-6*ABS(Y7
-.5):IF BB<AA THEN AA=BB
5030 CC=.5-2*ABS(X7-.5):DD=.25-ABS(Y7-.5
)
5040 IF CC>AA THEN AA=CC
5050 IF DD>AA THEN AA=DD
5060 Z=NP*13/48*AA
```

Modifications pour le dessin 195

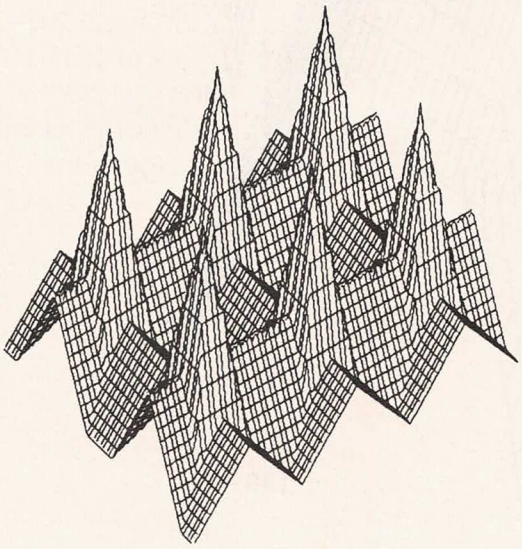
```
10 'SURFACES
100 N%=40:PA=NP/240:E1=2:E2=0
110 XA=NP*5/12:YA=0:XB=NP:YB=NP/4:XC=NP*
7/12:YC=NP/2:XD=0:YD=NP/4
5010 R2=SQR(2)
5020 AA=.5-ABS(X-.5):BB=.5-ABS(Y-.5):CC=
.5-ABS(X+Y-1)/R2:DD=.5-ABS(X-Y)/R2
5030 IF BB<AA THEN AA=BB
5040 IF CC<AA THEN AA=CC
5050 IF DD<AA THEN AA=DD
5060 Z=NP*5/2*AA
```

Modifications pour le dessin 196

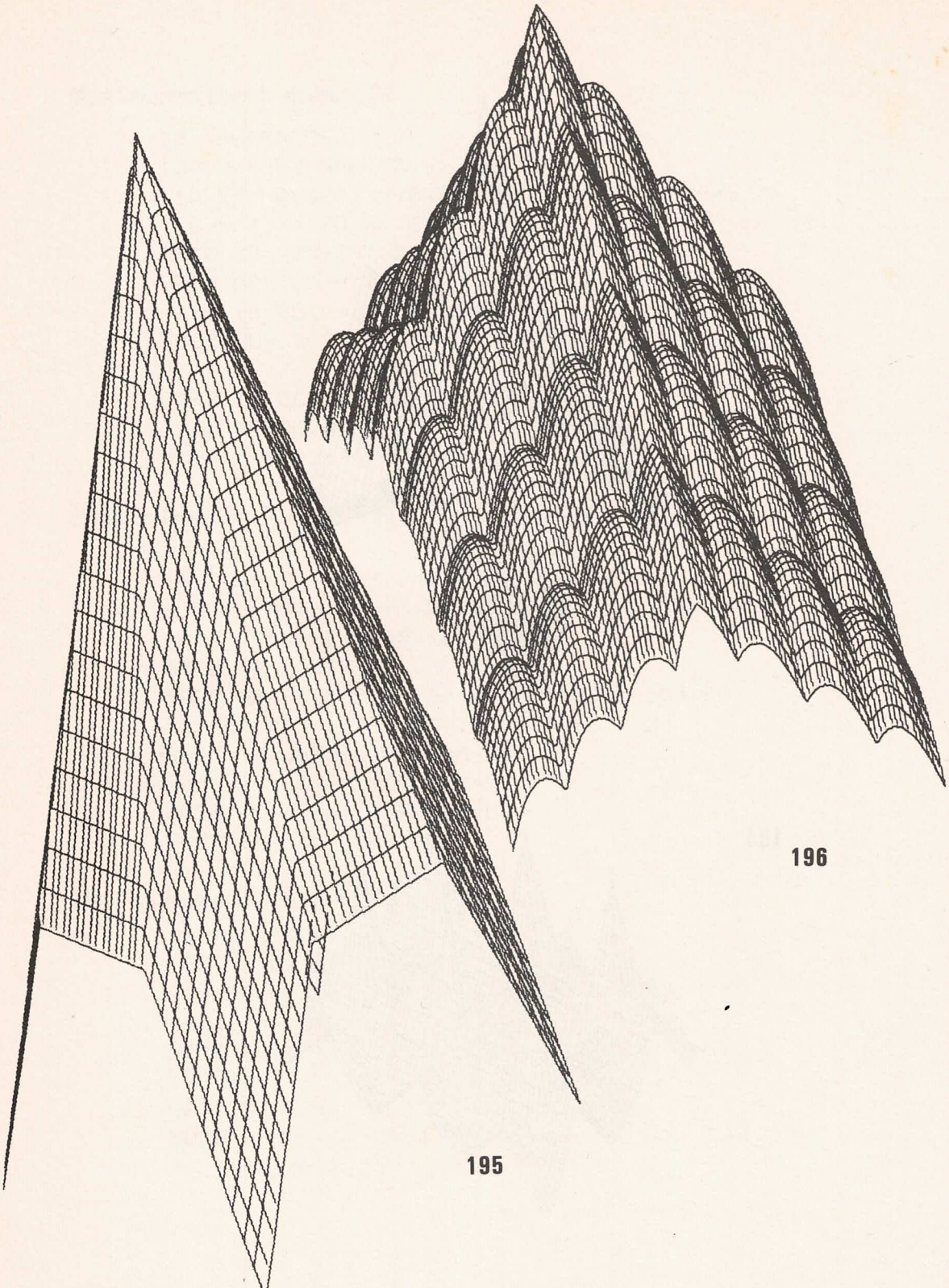
```
10 'SURFACES
100 N%=80:PA=NP/240:E1=2:E2=0
110 XA=NP/4:YA=0:XB=NP:YB=0:XC=NP*3/4:YC
=NP*3/4:XD=0:YD=NP*3/4
5010 X7=(7*X-INT(7*X))*2-1:Y7=(7*Y-INT(7
*Y))*2-1
5020 DI=X7*X7+Y7*Y7
5030 Z=NP*(1/16*(1-DI)+5/4*(1-ABS(X-.5))
*(1-ABS(Y-.5)))
```



193



194



196

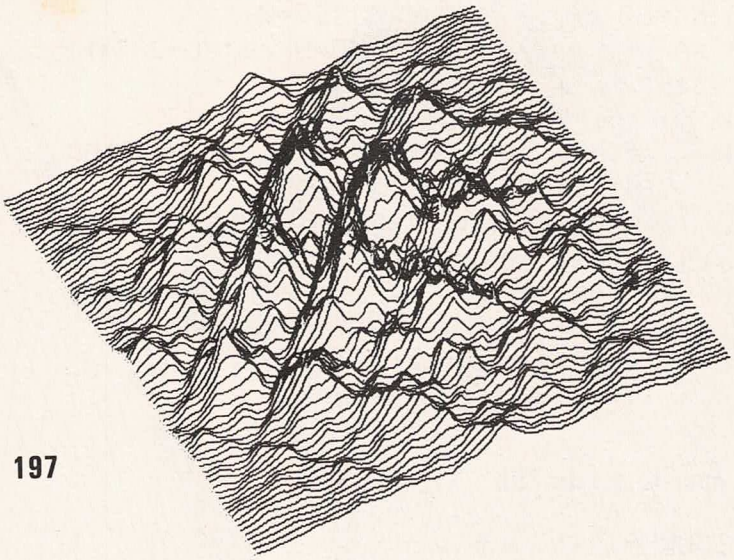
195

Modifications pour le dessin 197

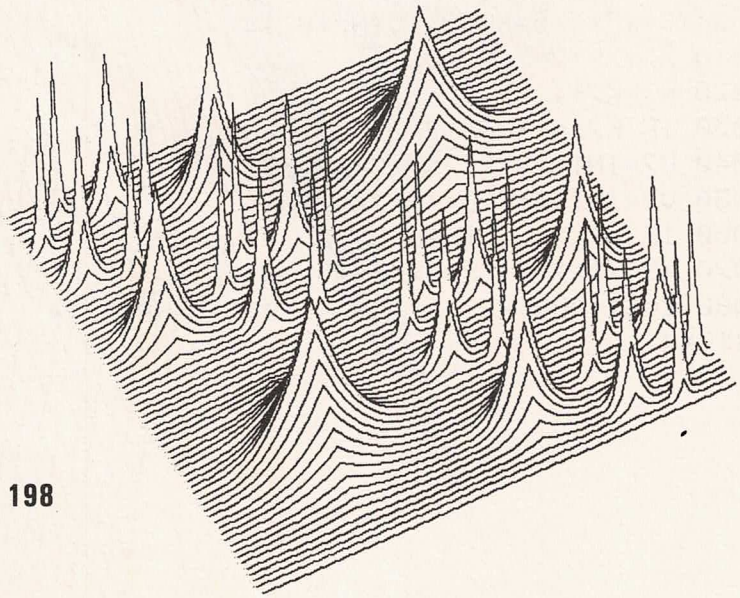
```
10 'SURFACES
100 N%=80:PA=NP/480:E1=1:E2=0
110 XA=NP/3:YA=0:XB=NP:YB=NP/4:XC=2/3*NP
:YC=3*NP/4:XD=0:YD=NP/2
5010 XH=X+.4755+Y:YH=Y+.23771-X
5020 X6=2*ABS(3*XH-INT(3*XH)-.5):Y6=2*AB
S(4*YH-INT(4*YH)-.5):Z6=X6*Y6
5030 X7=2*ABS(5*XH-INT(5*XH)-.5):Y7=2*AB
S(5*YH-INT(5*YH)-.5):Z7=X7*Y7
5040 X8=2*ABS(10*XH-INT(10*XH)-.5):Y8=2*
ABS(13*YH-INT(13*YH)-.5):Z8=X8*Y8
5050 Z=(5*Z6+3*Z7+2*Z8)*NP/48:Z=Z*(.6-AB
S(X-.5))*(.6-ABS(Y-.5))*4
```

Modifications pour le dessin 198

```
10 'SURFACES
100 N%=64:PA=NP/240:E1=1:E2=0
110 XA=NP/3:YA=0:XB=NP:YB=NP*4/15:XC=NP*
2/3:YC=NP*4/5:XD=0:YD=NP*8/15
5010 X7=X:Y7=Y:K7=0
5020 K7=K7+1
5030 IF K7>4 THEN Z=0: RETURN
5040 U7=INT(2*X7):X7=2*X7-U7
5050 V7=INT(2*Y7):Y7=2*Y7-V7
5060 IF U7<> V7 THEN 5020
5070 X9=2*X7-1:Y9=2*Y7-1
5080 Z=(1-ABS(X9))*(1-ABS(Y9)):Z=NP*5/24
*Z*Z*Z
```



197



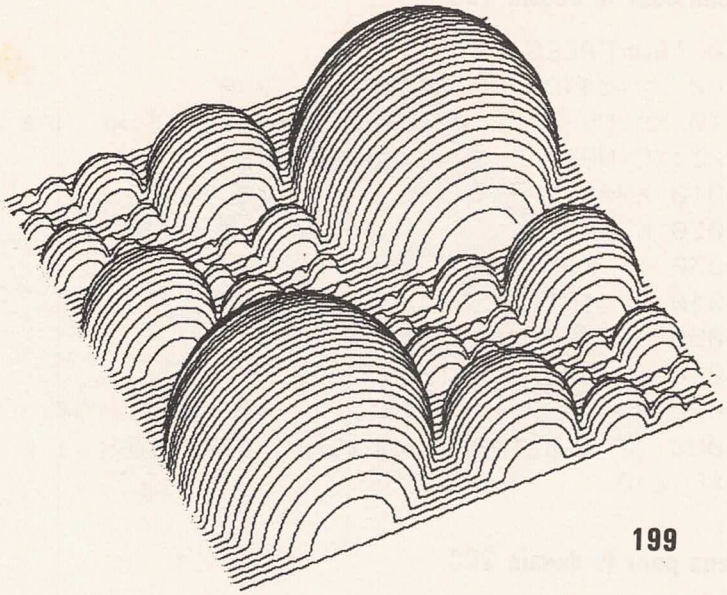
198

Modifications pour le dessin 199

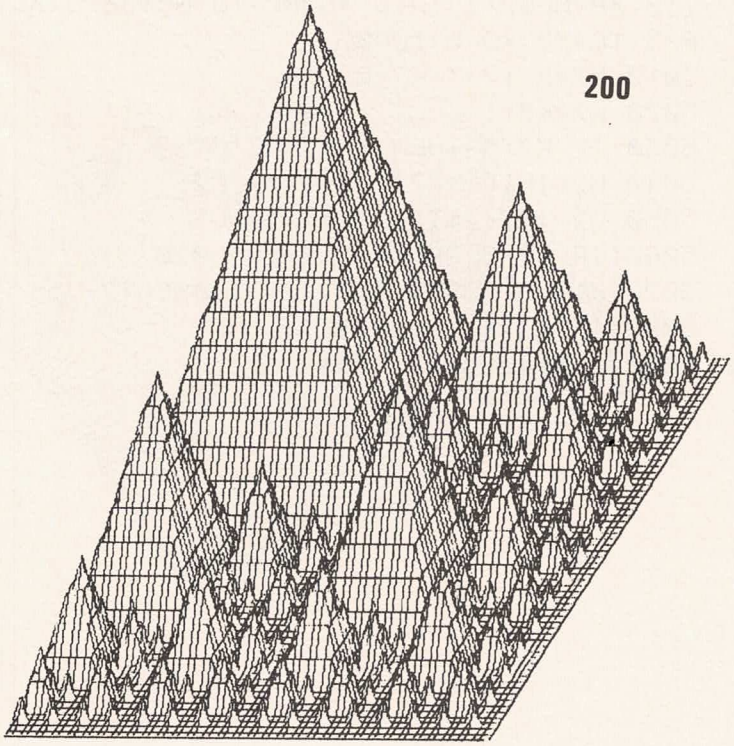
```
10 'SURFACES
100 N%=64:PA=NP/240:E1=1:E2=0
110 XA=NP/3:YA=0:XB=NP:YB=NP*4/15:XC=NP*
2/3:YC=NP*4/5:XD=0:YD=NP*8/15
5010 X7=X:Y7=Y:K7=0
5020 K7=K7+1
5030 IF K7>6 THEN Z=0: RETURN
5040 U7=INT(2*X7):X7=2*X7-U7
5050 V7=INT(2*Y7):Y7=2*Y7-V7
5060 IF U7<> V7 THEN 5020
5070 X9=2*X7-1:Y9=2*Y7-1:Z=1-X9*X9-Y9*Y9
5080 IF Z>0 THEN Z=NP*5/16/2^K7*SQR(Z) E
LSE Z=0
```

Modifications pour le dessin 200

```
10 'SURFACES
100 N%=64:PA=NP/240:E1=2:E2=0
110 XA=NP*2/3:YA=0:XB=NP:YB=NP*8/15:XC=N
P/3:YC=YB:XD=0:YD=0
5010 X7=X:Y7=Y:K7=0
5020 K7=K7+1
5030 IF K7>5 THEN Z=0: RETURN
5040 U7=INT(2*X7):X7=2*X7-U7
5050 V7=INT(2*Y7):Y7=2*Y7-V7
5060 IF U7=0 OR V7=0 THEN 5020
5070 Z1=.5-ABS(X7-.5):Z=.5-ABS(Y7-.5)
5080 IF Z1<Z THEN Z=Z1
5090 Z=NP*2.5/(2^K7)*Z
```



199



200

10 La troisième dimension

Pour terminer ce livre, nous proposons trois programmes permettant de réaliser des dessins d'objets de l'espace. Ils sont tous trois construits autour d'un sous-programme qui, à partir des coordonnées d'un point de l'espace, fournit les coordonnées du point correspondant de la feuille en tenant compte de la position de l'observateur (choisie au début du programme).

Pour réussir à utiliser le sous-programme PERSPECTIVE, il faut comprendre la manière dont on fixe les paramètres.

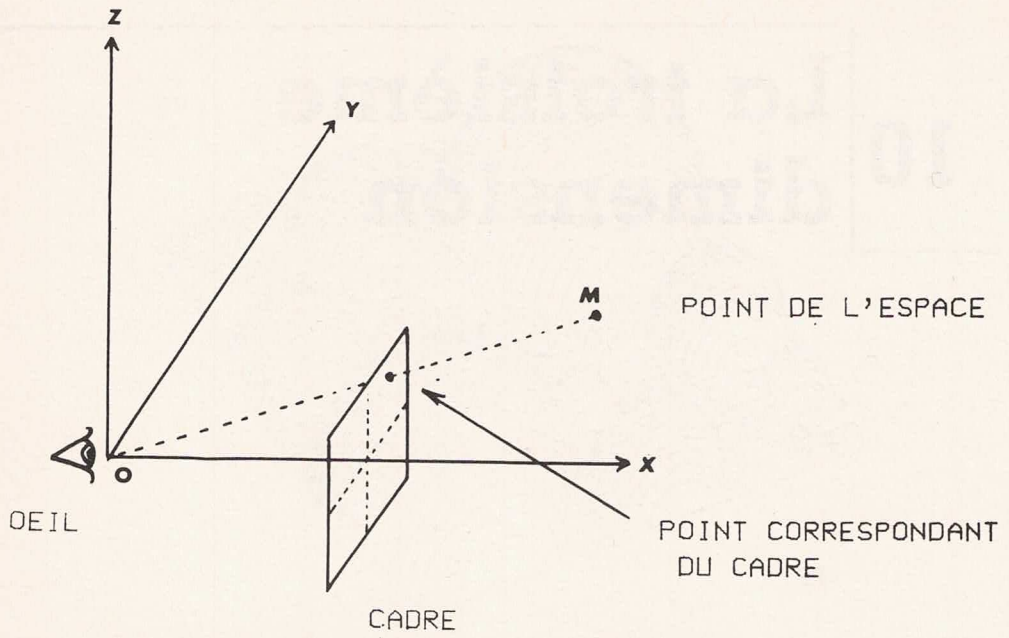
Pour cela, imaginons que nous sommes dans l'espace, à l'origine 0, et que nous regardons dans la direction de l'axe des x.

Nous avons devant nous un cadre transparent, carré, de taille TC, et situé à la distance DC (voir fig. p. 196).

A travers ce cadre, nous voyons une certaine partie de l'espace dont chaque point a son image sur le cadre (le point image d'un point M de l'espace s'obtient comme l'intersection de la droite OM avec le plan du cadre).

Si les coordonnées du point M sont MX, MY, MZ on calcule facilement que celles du point image sont données par :

KP = DC/MX
X = (— KP * MY/TC + .5) * NP
Y = (KP * MZ/TC + .5) * NP.



(Ces formules sont utilisées aux lignes 5 300 - 5 330).

Si maintenant, on souhaite savoir ce qui se passe lorsque notre système œil cadre est situé ailleurs dans l'espace, il suffit d'imaginer qu'on se déplace. Pour des raisons de commodité d'utilisation, nous considérons que successivement (l'ordre est important) :

— on se déplace au point de coordonnées OX, OY, OZ
— on pivote autour de l'axe Z d'un angle AZ
— on pivote autour de l'axe Y d'un angle AY
— on pivote autour de l'axe X d'un angle AX
— on se déplace à nouveau au point de coordonnées QX, QY, QZ.
(après chacun des déplacements le repère est changé et le déplacement suivant se fait relativement au nouveau repère).

Les formules qui correspondent à ces translations et rotations sont aux lignes 5 200 - 5 290.

Exemple 1

S'élever de 5 unités puis regarder vers le bas avec un angle de 45°, correspond à :

OX = Ø, OY = Ø, OZ = 5,
AZ = Ø, AY = —PI/4, AX = Ø,
QX = Ø, QY = Ø, QZ = Ø.

Exemple 2

Se déplacer à une distance de 10 unités de l'origine dans une direction faisant un angle de 37 ° avec l'axe de x, dans le plan xoy, et regarder vers l'origine correspond à :

OX = Ø, OY = Ø, OZ = Ø
AZ = 37 / 360 * 2 * PI, AY = Ø, AZ = Ø,
QX = — 10, QY = Ø, QZ = Ø.

REMARQUE.

Pour des raisons de simplicité et de rapidité d'exécution, nous n'effectuons pas de test de visibilité. Ceci a en particulier pour conséquence qu'un objet placé derrière l'observateur est projeté inversé sur le cadre.

Le programme D3DATA

Le programme qui suit permet de visualiser un petit édifice sous divers points de vue en effaçant les faces cachées (grâce à une astuce, et pourvu qu'on se trouve placé dans une certaine zone de l'espace), ou sans les effacer.

Ce programme est basé sur un principe de codification des suites de segments analogue à celui du chapitre 2. Les coordonnées en trois dimensions des segments qui composent l'édifice sont placées comme données des lignes 1 000 à 1 110 avec la ponctuation :

1 000 pour « changement de suite de segments » et
2 000 pour « fin de la liste des données » (voir les explications données au chapitre 2).

Ces coordonnées ne sont pas placées dans n'importe quel ordre, elles sont mises de telle façon que les segments décrits des lignes 1 000 à 1 060 correspondent aux segments visibles de l'édifice lorsqu'on se trouve dans la partie de l'espace X >0, Y < 0, Z >0.

En terminant la ligne 1 060 par 2 000, on aura donc une vue avec effacement des faces cachées (dessins 201, 202, 203) à condition d'être placé dans la bonne zone (ce que volontairement nous n'avons pas fait pour le dessin 205).

En terminant la ligne 1 060 par 5.5, on obtient une vision complète de l'édifice (et ceci bien sûr, quelle que soit la position dans l'espace) comme celle du dessin 204.

Ligne 50

Initialisation — Voir le chapitre 0.

Lignes 100 - 130

Fixation des paramètres du cadre et de la position de l'œil (les explications sont données en début de chapitre).

Lignes 200 - 220

Calcul des fonctions trigonométriques des angles AZ, AY, AX qui servent dans le sous-programme. On les calcule dès le début du programme pour ne pas avoir à les réévaluer à chaque appel du sous-programme.

Lignes 1 000 - 1 110

Données définissant l'édifice à représenter, codées selon le principe expliqué plus haut.

Lignes 2 000 - 2 100

Partie principale du programme, construite sur le modèle du premier programme du chapitre 2.

Lignes 5 000 à 5 500

Sous-programme de calcul et de tracé des segments en trois dimensions. A partir de MX, MY, MZ (en utilisant les paramètres de vision placés en début de programme) ce sous-programme

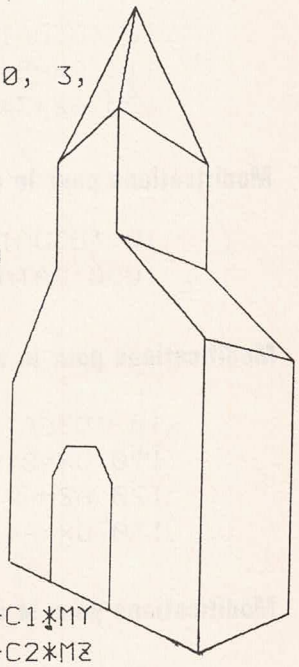
- trace un segment entre le point où est la plume et le point image de (MX, MY, MZ) si B1 = 1
- se déplace, plume levée, au point image de (MX, MY, MZ) si B1 = 0.

Les calculs sont expliqués au début du chapitre.

Programme de base « D3 DATA »

```
10 'D3DATA
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 DC=2:TC=2
110 OX=5:OY=-2:OZ=1.3
120 AZ=3*PI/4:AY=0:AX=0
130 QX=0:QY=0:QZ=0
200 C1=COS(AZ):S1=-SIN(AZ)
```

```
210 C2=COS(AY):S2=-SIN(AY)
220 C3=COS(AX):S3=-SIN(AX)
1000 DATA 0,0,0, 0,0,2, 1,0,3, 1,0,4, 2,
0,4, 2,0,3, 3,0,2, 3,0,0, 3,1,0, 3,1,2
1010 DATA 2,1,3, 2,1,4, 2,0,4, 1.5,.5,5.
5, 1,0,4, 1000
1020 DATA 0,0,0, 3,0,0, 1000
1030 DATA 1,0,0, 1,0,1, 1.25,0,1.3, 1.75
,0,1.3, 2,0,1, 2,0,0, 1000
1040 DATA 3,0,2, 3,1,2, 1000
1050 DATA 2,0,3, 2,1,3, 1000
1060 DATA 2,1,4, 1.5,.5,5.5, 2000
1070 DATA 1,1,4, 1,1,3, 0,1,2, 0,1,0, 3,
1,0, 1000
1080 DATA 0,0,0, 0,1,0, 1000
1090 DATA 0,0,2, 0,1,2, 1000
1100 DATA 1,0,3, 1,1,3, 1000
1110 DATA 1,0,4, 1,1,4, 2,1,4, 2000
2000 READ MX
2010 IF MX=2000 THEN 2100
2020 IF MX=1000 THEN B1=0:READ MX
2030 READ MY,MZ
2040 GOSUB 5000
2050 GOTO 2000
2100 END
5000 REM SOUS-PROG PERSPECTIVE
5200 MX=MX-OX:MY=MY-OY:MZ=MZ-OZ
5250 UU=MX:MX=C1*UU-S1*MY:MY=S1*UU+C1*MY
5260 UU=MX:MX=C2*UU-S2*MZ:MZ=S2*UU+C2*MZ
5270 UU=MY:MY=C3*UU-S3*MZ:MZ=S3*UU+C3*MZ
5290 MX=MX-QX:MY=MY-QY:MZ=MZ-QZ
5300 KP=DC/MX
5310 XX=-KP*MY:YY=KP*MZ
5320 X%=INT(NP*(.5+XX/TC))
5330 Y%=INT(NP*(.5+YY/TC))
5400 IF B1=1 THEN LPRINT"D";X%;",";Y%
5410 IF B1=0 THEN B1=1:LPRINT"M";X%;",";
Y%
5500 RETURN
```



Modifications pour le dessin 201

```
10 'D3DATA
```

Modifications pour le dessin 202

```
10 'D3DATA
110 OX=5:OY=-2:OZ=6
120 AZ=3*PI/4:AY=-PI/4:AX=0
```

Modifications pour le dessin 203

```
10 'D3DATA
110 OX=5:OY=-3:OZ=0
120 AZ=3*PI/4:AY=PI/4:AX=0
```

Modifications pour le dessin 204

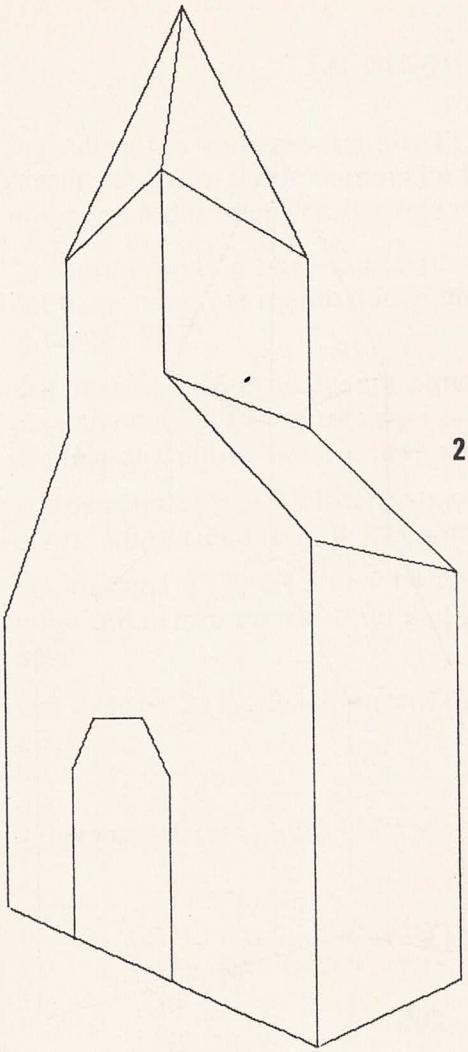
```
10 'D3DATA
1060 DATA 2,1,4, 1.5,.5,5.5
```

Modifications pour le dessin 205

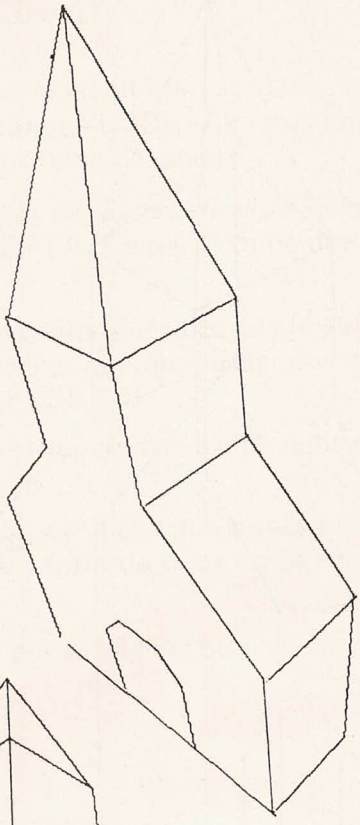
```
10 'D3DATA
110 OX=0:OY=0:OZ=1.3
120 AZ=-PI/4:AY=0:AX=0
130 QX=-4:QY=1:QZ=0
```

Modifications pour le dessin 206

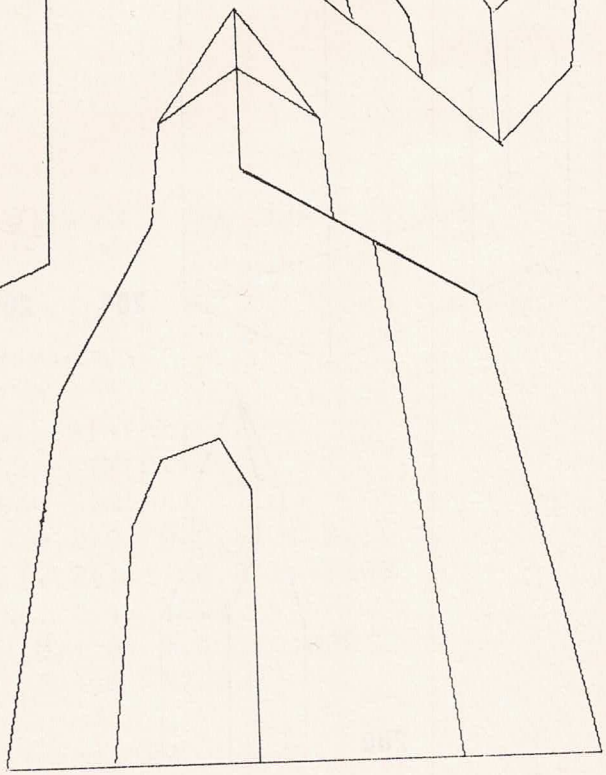
```
10 'D3DATA
110 OX=12:OY=-4:OZ=1.3
120 AZ=.78*PI:AY=0:AX=0
1900 FOR I=0 TO 2
2010 IF MX=2000 THEN GOTO 2100
2030 READ MY,MZ :MY=MY+6*I
2100 RESTORE:B1=0:NEXT I
2200 END
```



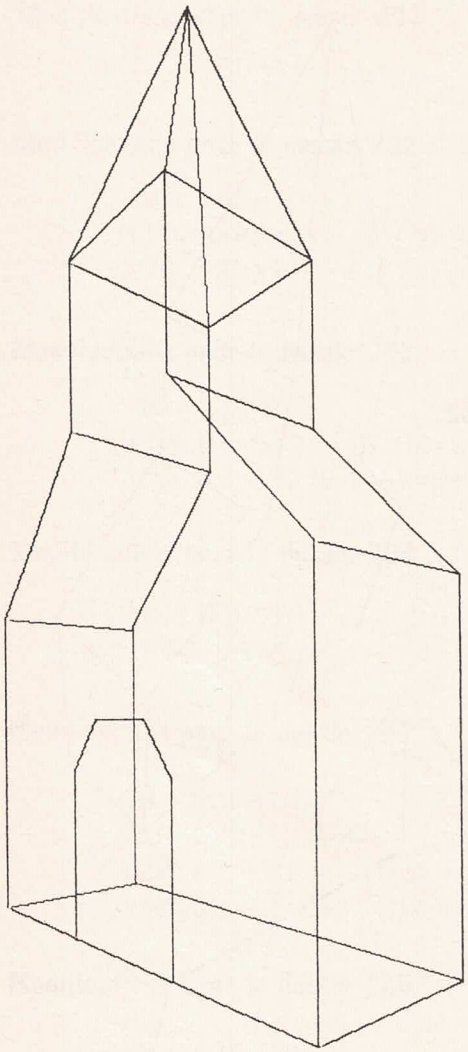
201



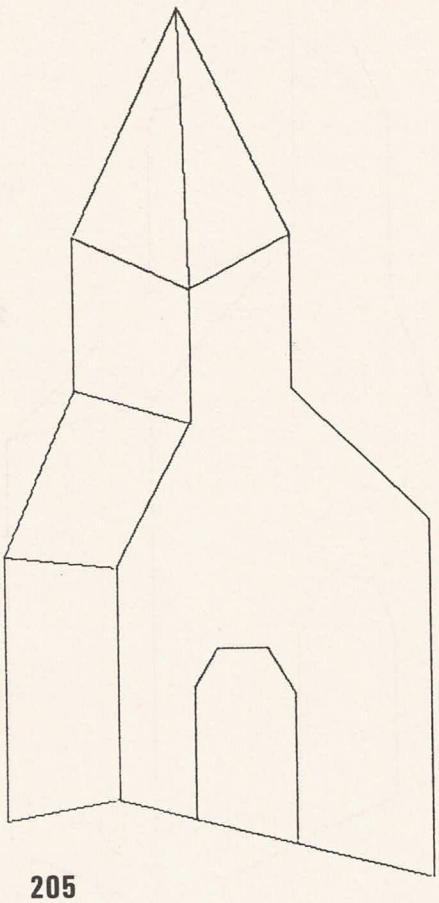
202



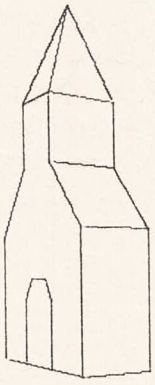
203



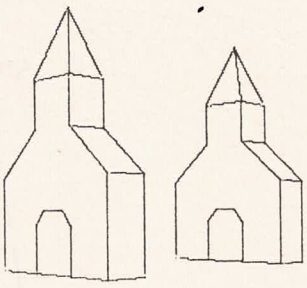
204



205



206



Le programme D3CUBE

En utilisant le sous-programme PERSPECTIVE et une suite de données définissant un cube (codé comme l'édifice du programme D3DATA), nous nous amusons à dessiner des familles de cubes placés dans l'espace.

Le programme de base (mis à part les données et les paramètres de vision) est exactement le même que le programme D3DATA. Ce programme donne le dessin 207.

Les dessins 208 et 209 sont obtenus en représentant une suite de cubes de plus en plus petits, centrés sur l'origine. Les formules qui font passer du cube modèle aux cubes transformés sont à la ligne 2 035.

Les dessins 210 et 211 correspondent à des vues superposées d'un cube qui pivote autour de l'une de ses arêtes.

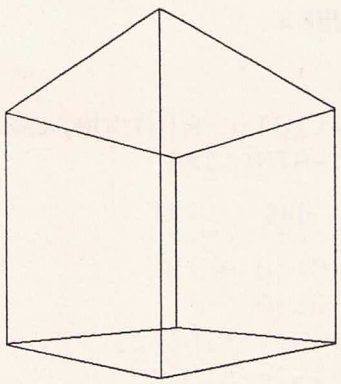
Les dessins 212, 213 et 214 correspondent à des vues superposées d'un cube qui pivote autour d'un axe passant par le centre de deux faces opposées.

Les dessins 215-220 montrent des familles de cubes alignés ou disposés en cube.

Programme de base « D3 CUBE »

```
10 'D3CUBE
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 DC=2:TC=2
110 OX=1.9:OY=-.8:OZ=.25
120 AZ=3*PI/4:AY=0:AX=0
130 QX=0:QY=0:QZ=0
200 C1=COS(AZ):S1=-SIN(AZ)
210 C2=COS(AY):S2=-SIN(AY)
220 C3=COS(AX):S3=-SIN(AX)
1000 DATA 1,0,0, 0,0,0, 0,0,1, 1,0,1, 1,
0,0, 1,1,0, 1,1,1, 0,1,1, 0,0,1, 1000
1010 DATA 1,0,1, 1,1,1, 1000
1020 DATA 0,0,0, 0,1,0, 0,1,1, 1000
1030 DATA 1,1,0, 0,1,0, 2000
2000 READ MX
2010 IF MX=2000 THEN 2100
```

```
2020 IF MX=1000 THEN B1=0:READ MX
2030 READ MY,MZ
2040 GOSUB 5000
2050 GOTO 2000
2100 END
5000 REM SOUS-PROG PERSPECTIVE
5200 MX=MX-OX:MY=MY-OY:MZ=MZ-OZ
5250 UU=MX:MX=C1*UU-S1*MY:MY=S1*UU+C1*MY
5260 UU=MX:MX=C2*UU-S2*MZ:MZ=S2*UU+C2*MZ
5270 UU=MY:MY=C3*UU-S3*MZ:MZ=S3*UU+C3*MZ
5290 MX=MX-QX:MY=MY-QY:MZ=MZ-QZ
5300 KP=DC/MX
5310 XX=-KP*MY:YY=KP*MZ
5320 X%=INT(NP*(.5+XX/TC))
5330 Y%=INT(NP*(.5+YY/TC))
5400 IF B1=1 THEN LPRINT"D";X%;",";Y%
5410 IF B1=0 THEN B1=1:LPRINT"M";X%;",";
Y%
5500 RETURN
```



Modifications pour le dessin 207

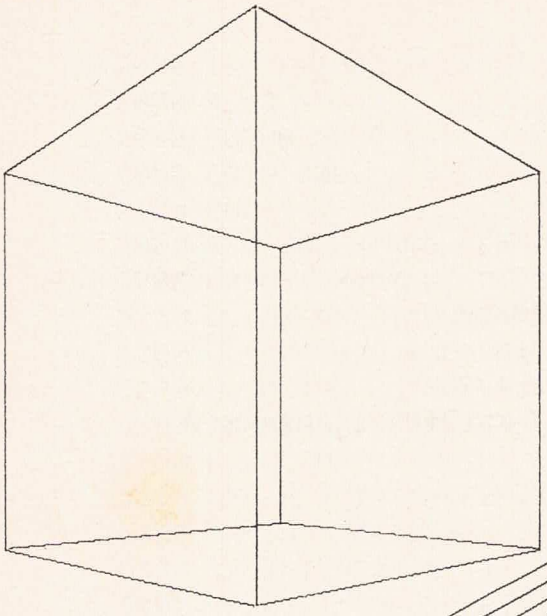
```
10 'D3CUBE
```

Modifications pour le dessin 208

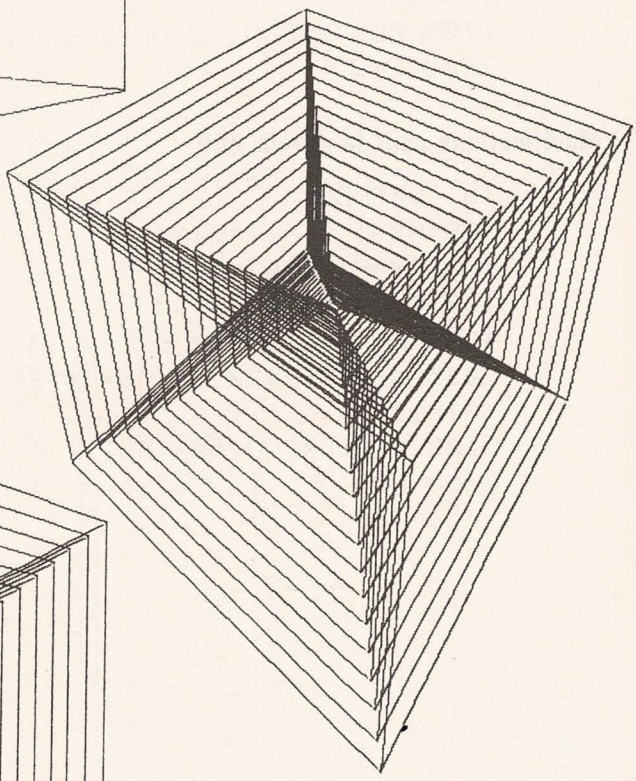
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/5:AY=-PI/7:AX=0
130 QX=-3.5:QY=0:QZ=1.5
1900 FOR I=1 TO20:K=1-I/20
2035 MX=K*(2*MX-1):MY=K*(2*MY-1):MZ=K*(2
*MZ-1)
2100 RESTORE:B1=0:NEXT I
2200 END
```

Modifications pour le dessin 209

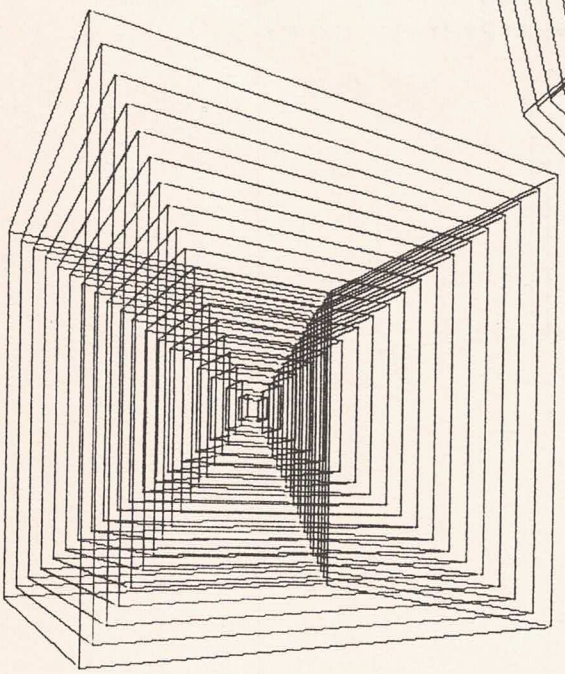
```
10 'D3CUBE
110 OX=1:OY=-1:OZ=-.5
120 AZ=PI/3:AY=0:AX=0
130 QX=-3:QY=1:QZ=0
1900 FOR I=1 TO20:K=1-I/20
2035 MX=K*(2*MX-1):MY=K*(2*MY-1):MZ=K*(2
*MZ-1)
2100 RESTORE:B1=0:NEXT I
2200 END
```



207



208



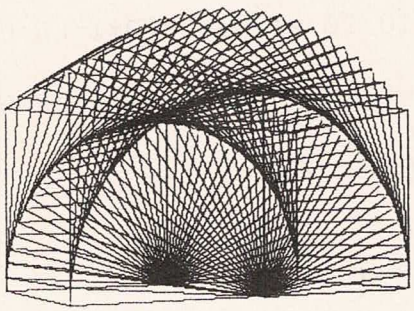
209

Modifications pour le dessin 210

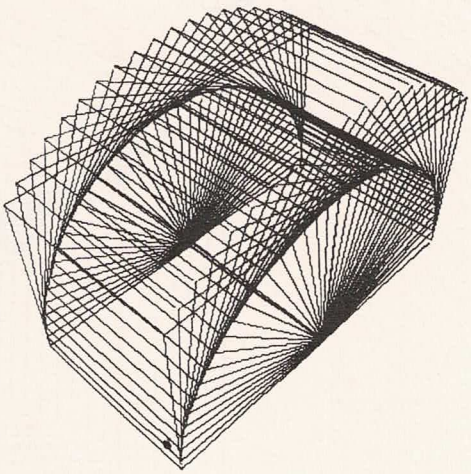
```
10 'D3CUBE
110 OX= 0:OY=.5:OZ=0
120 AZ=PI/3:AY=0:AX=0
130 QX=-4:QY=.1:QZ=.3
1900 FOR I=0 TO 20:AT=PI/2/20*I:CC=COS(A
T):SS=SIN(AT)
2035 GR=MX:MX=CC*MX-SS*MZ:MY=MY:MZ=GR*SS
+MZ*CC
2100 RESTORE:B1=0:NEXT I
2200 END
```

Modifications pour le dessin 211

```
10 'D3CUBE
110 OX= 0:OY=.5:OZ=0
120 AZ=PI/4:AY=-PI/4:AX=0
130 QX=-4:QY=.1:QZ=.3
1900 FOR I=0 TO 20:AT=PI/2/20*I:CC=COS(A
T):SS=SIN(AT)
2035 GR=MX:MX=CC*MX-SS*MZ:MY=MY:MZ=GR*SS
+MZ*CC
2100 RESTORE:B1=0:NEXT I
2200 END
```



210



211

Modifications pour le dessin 212

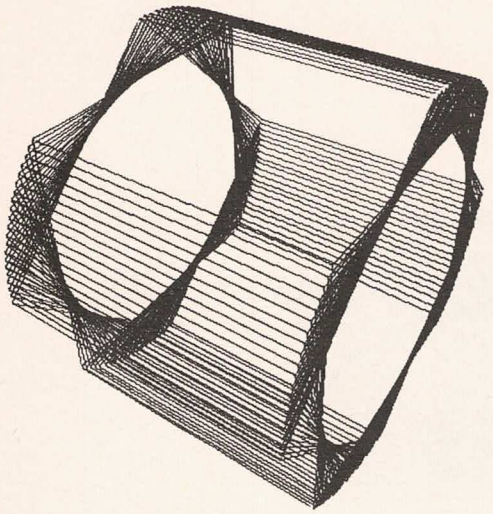
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/5:AY=-PI/5:AX=0
130 QX=-5:QY=.2:QZ=.3
1900 FOR I=0 TO 20:AT=PI/2/40*I:CC=COS(A
T):SS=SIN(AT)
2035 GR=2*MX-1:MX=CC*GR-SS*(2*MZ-1):MY=2
*MY-1:MZ=GR*SS+(2*MZ-1)*CC
2100 RESTORE:B1=0:NEXT I
2200 END
```

Modifications pour le dessin 213

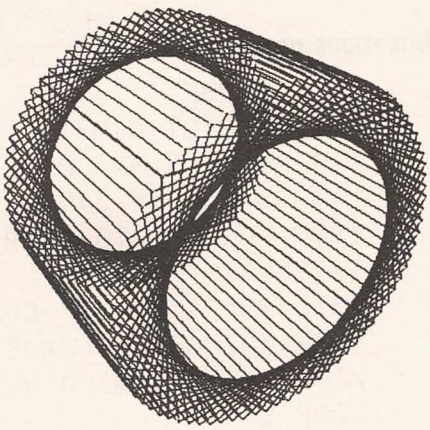
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/3:AY=-PI/7:AX=0
130 QX=-6:QY=.3:QZ=.5
1900 FOR I=0 TO 20:AT=PI/2/20*I:CC=COS(A
T):SS=SIN(AT)
2035 GR=2*MX-1:MX=CC*GR-SS*(2*MZ-1):MY=2
*MY-1:MZ=GR*SS+(2*MZ-1)*CC
2100 RESTORE:B1=0:NEXT I
2200 END
```

Modifications pour le dessin 214

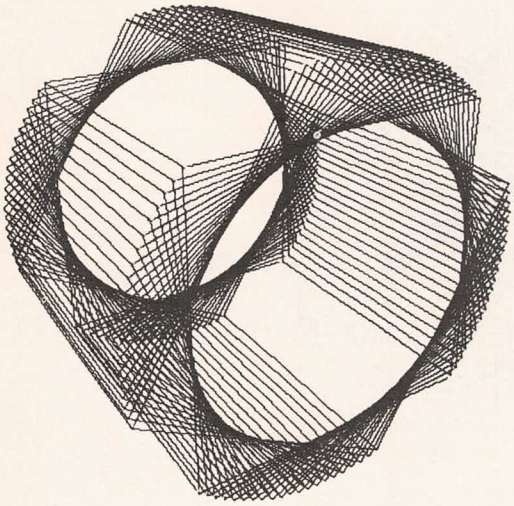
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/3:AY=-PI/8:AX=0
130 QX=-5:QY=.2:QZ=.3
1900 FOR I=0 TO 20:AT=PI/2/30*I:CC=COS(A
T):SS=SIN(AT)
2035 GR=2*MX-1:MX=CC*GR-SS*(2*MZ-1):MY=2
*MY-1:MZ=GR*SS+(2*MZ-1)*CC
2100 RESTORE:B1=0:NEXT I
2200 END
```



212



213



214

Modifications pour le dessin 215

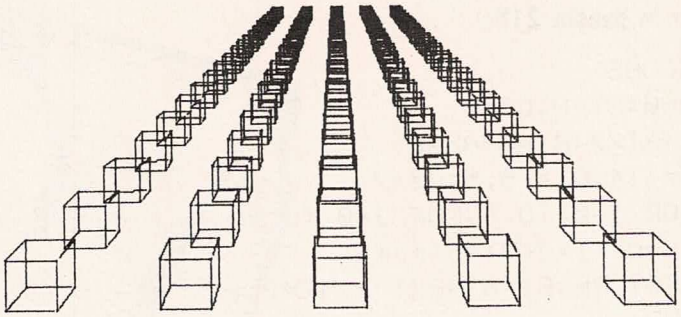
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/2:AY=0:AX=0
130 QX=-14:QY=-6.5:QZ=7
1900 FOR I=0 TO 4:FOR J=0 TO 20
2035 MX=MX+I*3:MY=MY+J*3
2100 RESTORE:B1=0:NEXT J:NEXT I
2200 END
```

Modifications pour le dessin 216

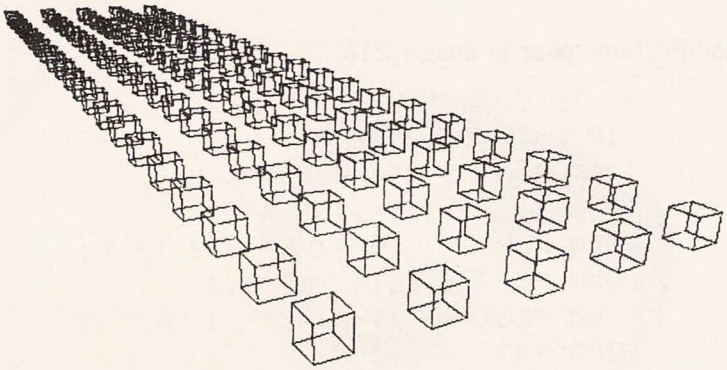
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/3:AY=-PI/8:AX=0
130 QX=-15:QY=-1:QZ=3
1900 FOR I=0 TO 4:FOR J=0 TO 20
2035 MX=MX+I*3:MY=MY+J*3
2100 RESTORE:B1=0:NEXT J:NEXT I
2200 END
```

Modifications pour le dessin 217

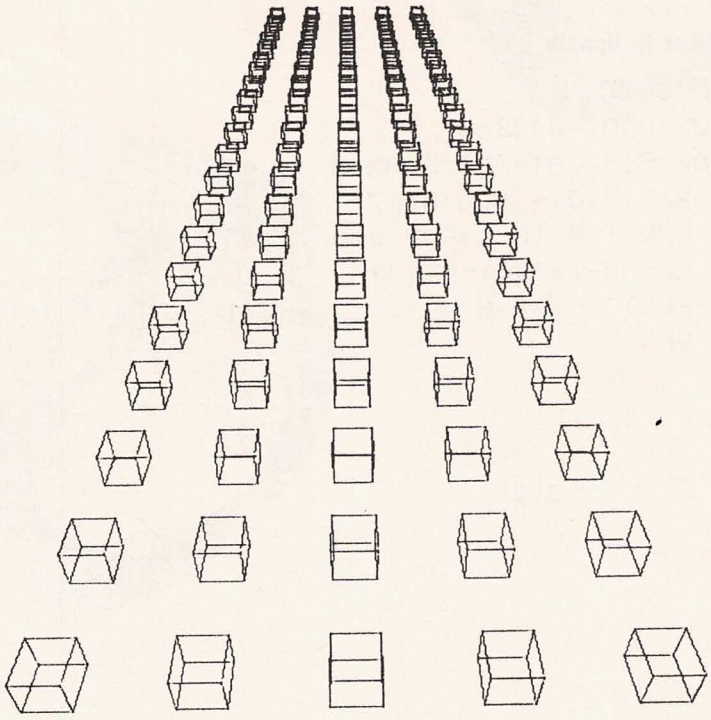
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/2:AY=-PI/5:AX=0
130 QX=-14:QY=-6.5:QZ=7
1900 FOR I=0 TO 4:FOR J=0 TO 20
2035 MX=MX+I*3:MY=MY+J*3
2100 RESTORE:B1=0:NEXT J:NEXT I
2200 END
```



215



216



217

Modifications pour le dessin 218

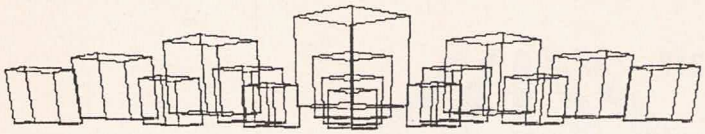
```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/4:AY=-PI/8:AX=0
130 QX=-12:QY=0:QZ=4.3
1900 FOR I=0 TO 3:FOR J=0 TO 3:FOR K=0 T
O 3
2035 MX=MX+I*3:MY=MY+J*3:MZ=MZ+K*3
2100 RESTORE:B1=0:NEXT K:NEXT J:NEXT I
2200 END
```

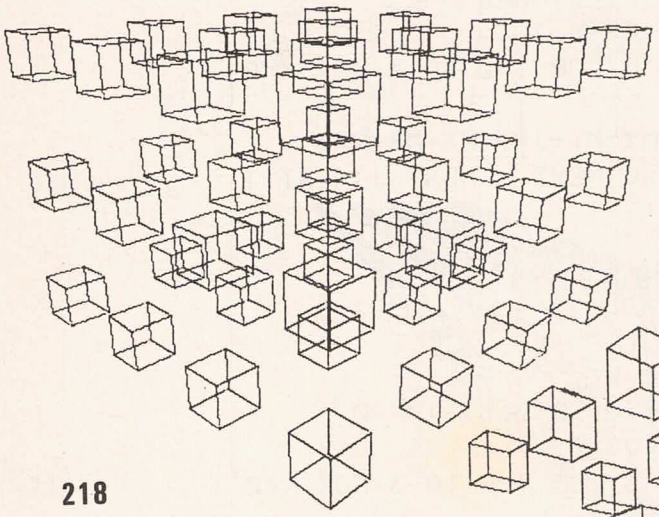
Modifications pour le dessin 219

```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/4:AY=ATN(SQR(.5)):AX=0
130 QX=-10:QY=0:QZ=0
1900 FOR I=0 TO 3:FOR J=0 TO 3:FOR K=0 T
O 3
2035 MX=MX+I*3:MY=MY+J*3:MZ=MZ+K*3
2100 RESTORE:B1=0:NEXT K:NEXT J:NEXT I
2200 END
```

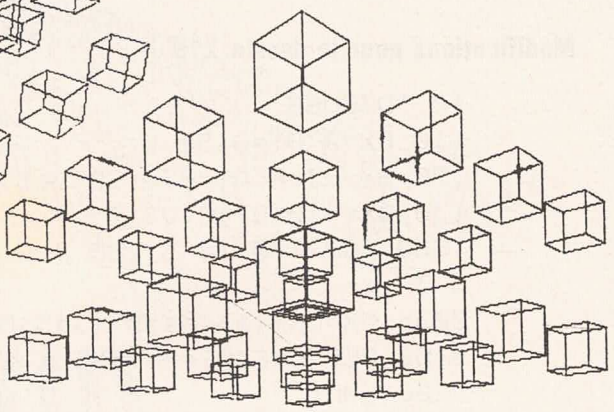
Modifications pour le dessin 220

```
10 'D3CUBE
110 OX=0:OY=0:OZ=0
120 AZ=PI/4:AY=0:AX=0
130 QX=-10:QY=0:QZ=3.5
1900 FOR I=0 TO 3:FOR J=0 TO 3:FOR K=0 T
O 3
2035 MX=MX+I*3:MY=MY+J*3:MZ=MZ+K*3
2100 RESTORE:B1=0:NEXT K:NEXT J:NEXT I
2200 END
```

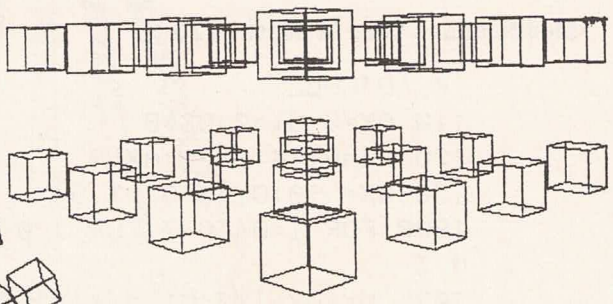


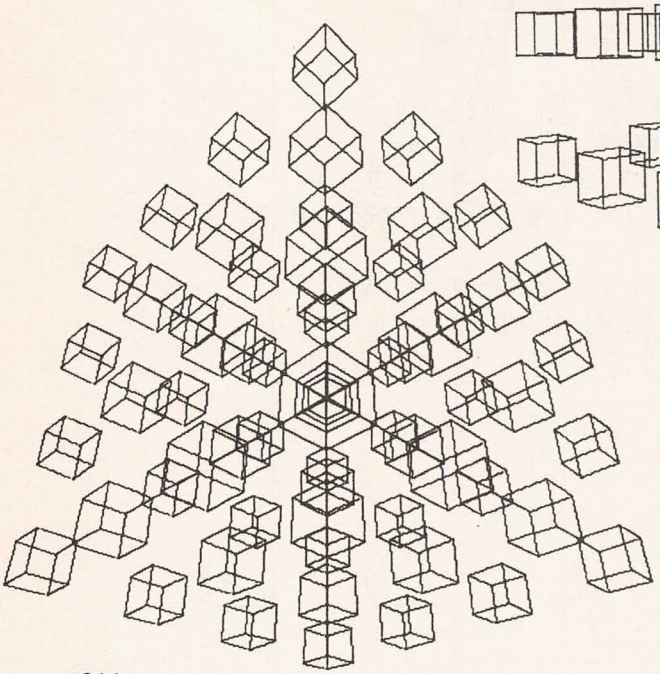


218



219





220

214

Le programme D3STRUCTURES

Ce programme permet de dessiner une série de sphères emboîtées les unes dans les autres (dessins 221, 234, 235) ainsi que divers objets tridimensionnels sous n'importe quel angle de vue. Les sphères sont obtenues en représentant la trajectoire d'un point qui se déplace à la surface d'une boule. On obtient les dessins 224-233 en joignant des points assez éloignés de cette trajectoire. Les paramètres de cette première version du programme sont à la ligne 2 000 (il y a bien sûr en plus les paramètres de vision) :

N : nombre de points sur la courbe.

M : nombre de doubles tours effectués par la courbe autour du centre de la sphère.

K + 1 : nombre de sphères.

Les dessins à partir du dessin 236 sont obtenus en changeant toutes les lignes de 2 000 à 2 500.

Ces divers exemples que nous vous laissons analyser montrent que l'ordinateur permet, sans utiliser ni marbre ni ciseaux, de « sculpter » des formes complexes de l'espace. A vous d'en imaginer d'autres et de voyager autour.

Programme de base « D 3 STRUCTURES »

```
10 'D3STRUCTURES
50 LPRINT CHR$(18):LPRINT"M0,-500":LPRIN
T"I":NP=480:PI=ATN(1)*4
100 DC=2:TC=2
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-PI/5:AX=0
130 QX=-2.5:QY=0:QZ=0
200 C1=COS(AZ):S1=-SIN(AZ)
210 C2=COS(AY):S2=-SIN(AY)
220 C3=COS(AX):S3=-SIN(AX)
2000 N=2000:M=30:K=0
2100 FOR J=0 TO K:RA=.6^J:B1=0
2110 FOR I=0 TO N
2120 AA=2*PI*I/N
2130 AB=AA*M
2140 CA=COS(AA):SA=SIN(AA):CB=COS(AB):SB
=SIN(AB)
```

```
2150 MX=RA*CA*CB
2160 MY=RA*SA*CB
2170 MZ=RA*SB
2200 GOSUB 5000
2300 NEXT I:NEXT J
2500 END
5000 REM SOUS-PROG PERSPECTIVE
5200 MX=MX-OX:MY=MY-OY:MZ=MZ-OZ
5250 UU=MX:MX=C1*UU-S1*MY:MY=S1*UU+C1*MY
5260 UU=MX:MX=C2*UU-S2*MZ:MZ=S2*UU+C2*MZ
5270 UU=MY:MY=C3*UU-S3*MZ:MZ=S3*UU+C3*MZ
5290 MX=MX-QX:MY=MY-QY:MZ=MZ-QZ
5300 KP=DC/MX
5310 XX=-KP*MY:YY=KP*MZ
5320 X%=INT(NP*(.5+XX/TC))
5330 Y%=INT(NP*(.5+YY/TC))
5400 IF B1=1 THEN LPRINT"D";X%;",";Y%
5410 IF B1=0 THEN B1=1:LPRINT"M";X%;",";
Y%
5500 RETURN
```

Modifications pour le dessin 221

```
10 'D3STRUCTURES
```

Modifications pour le dessin 222

```
10 'D3STRUCTURES
2000 N=200:M=4:K=0
```

Modifications pour le dessin 223

```
10 'D3STRUCTURES
2000 N=400:M=7:K=0
```

Modifications pour le dessin 224

```
 0 'D3STRUCTURES
120 AZ=0:AY=-PI/5:AX=0
2000 N=400:M=100:K=0
```

Modifications pour le dessin 225

```
10 'D3STRUCTURES
120 AZ=0:AY=-2*PI/5:AX=0
2000 N=400:M=100:K=0
```

Modifications pour le dessin 226

```
10 'D3STRUCTURES
120 AZ=0:AY=-PI/5:AX=0
2000 N=120:M=30:K=8
2100 FOR J=0 TO K:RA=.8^J:B1=0
```

Modifications pour le dessin 227

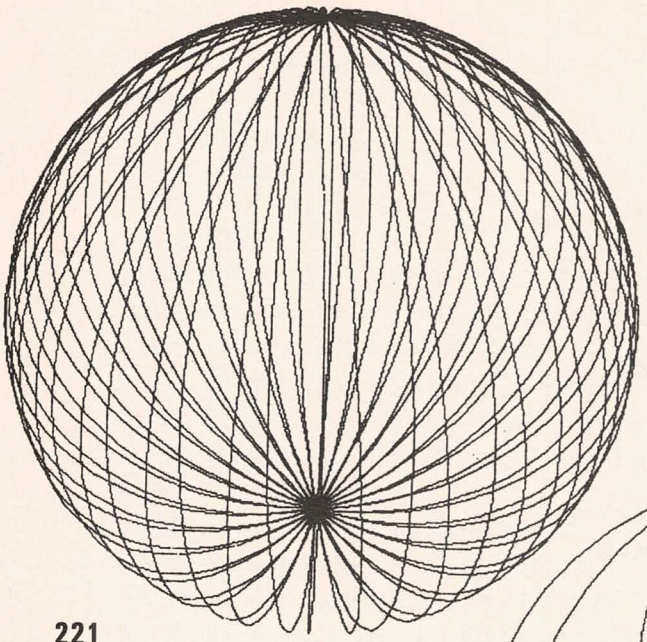
```
10 'D3STRUCTURES
120 AZ=0:AY=-2*PI/5:AX=0
2000 N=120:M=30:K=8
2100 FOR J=0 TO K:RA=.8^J:B1=0
```

Modifications pour le dessin 228

```
10 'D3STRUCTURES
2000 N=300:M=100:K=0
```

Modifications pour le dessin 229

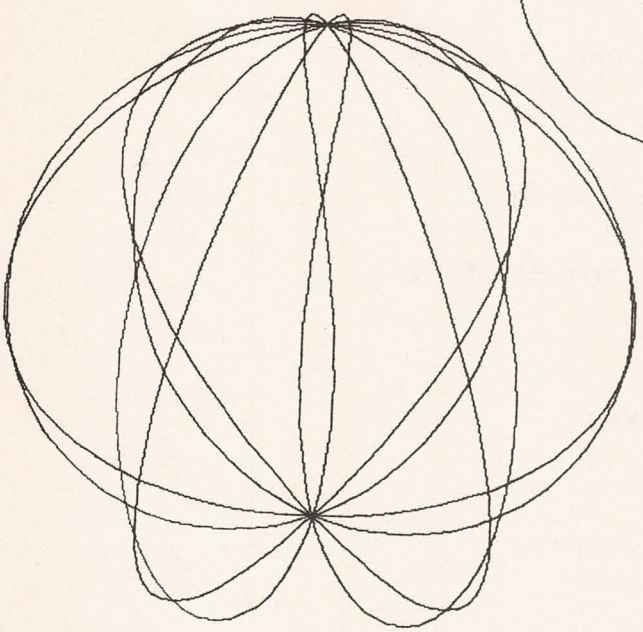
```
10 'D3STRUCTURES
120 AZ=0:AY=-2*PI/5:AX=0
2000 N=300:M=100:K=0
```

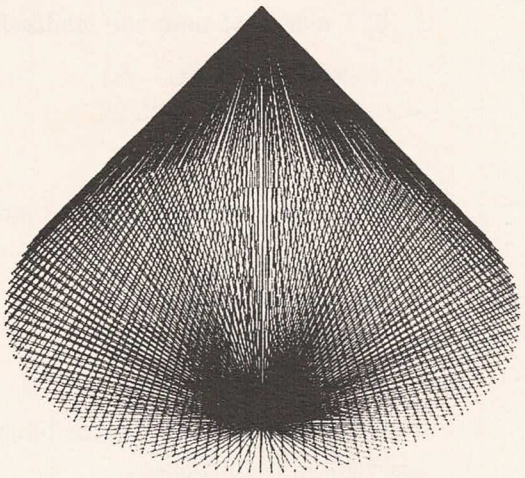


221

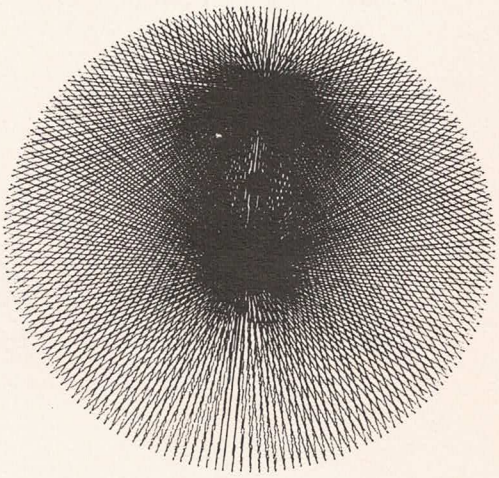
222

223

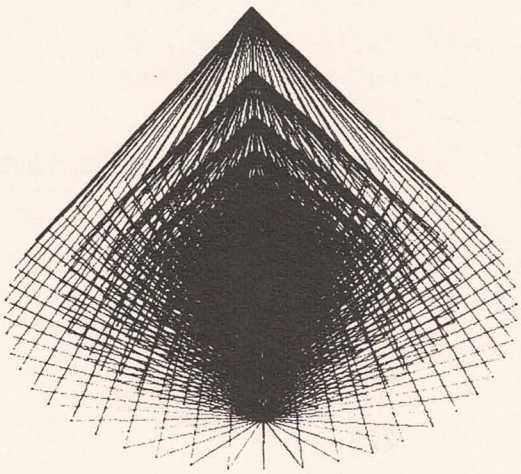




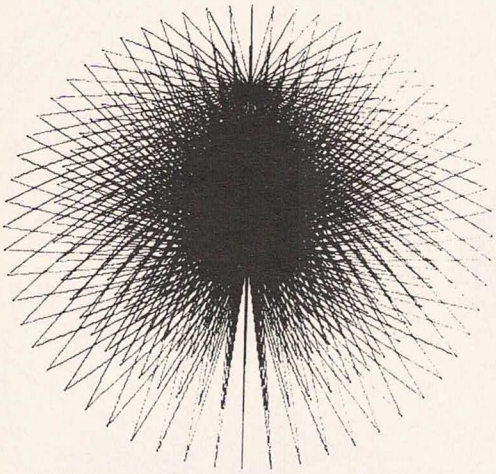
224



225

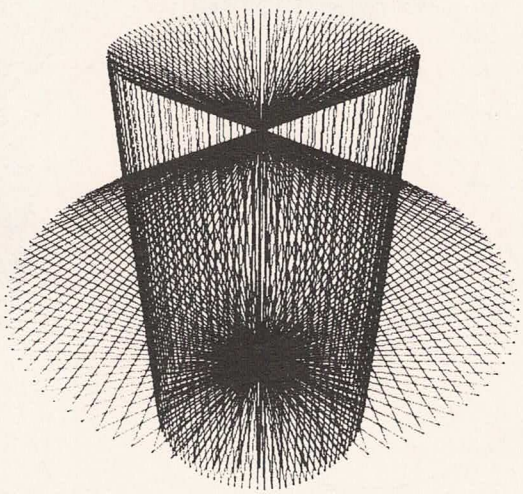


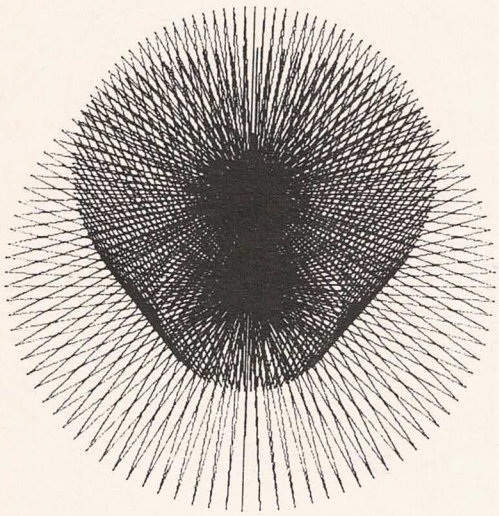
226



227

228





229

Modifications pour le dessin 230

```
10 'D3STRUCTURES
2000 N=300:M=60:K=0
```

Modifications pour le dessin 231

```
10 'D3STRUCTURES
120 AZ=0:AY=-2*PI/5:AX=0
2000 N=300:M=60:K=0
```

Modifications pour le dessin 232

```
10 'D3STRUCTURES
120 AZ=0:AY=-PI/5:AX=0
2000 N=600:M=100:K=0
```

Modifications pour le dessin 233

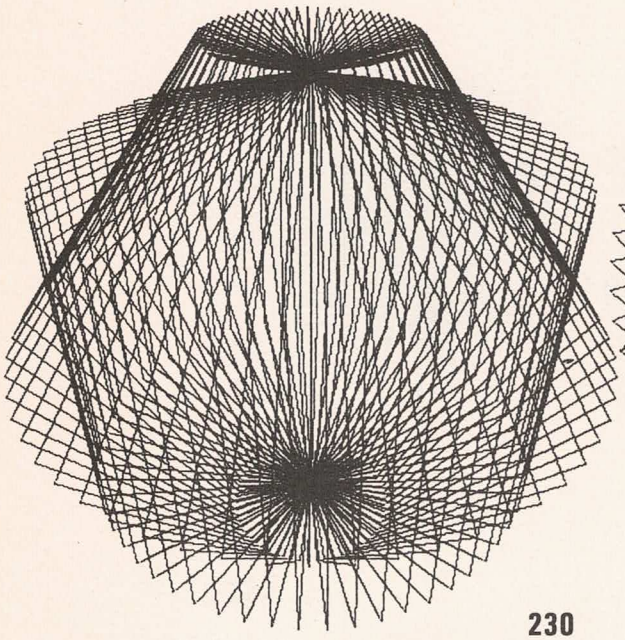
```
10 'D3STRUCTURES
120 AZ=0:AY=-2*PI/5:AX=0
2000 N=600:M=100:K=0
```

Modifications pour le dessin 234

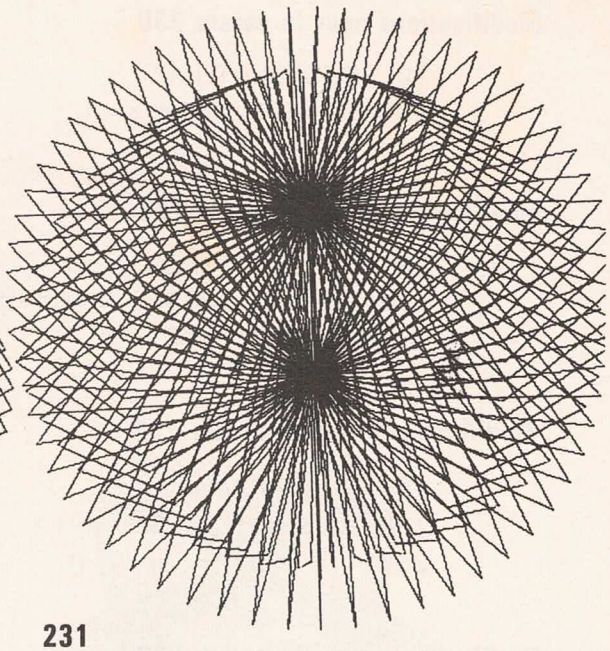
```
10 'D3STRUCTURES
120 AZ=0:AY=-PI/5:AX=0
2000 N=1000:M=20:K=3
```

Modifications pour le dessin 235

```
10 'D3STRUCTURES
120 AZ=0:AY=-2*PI/5:AX=0
2000 N=1000:M=20:K=3
```

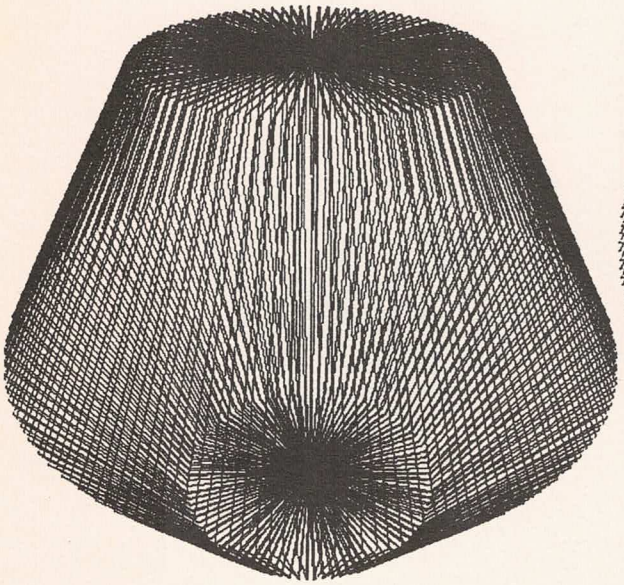


230

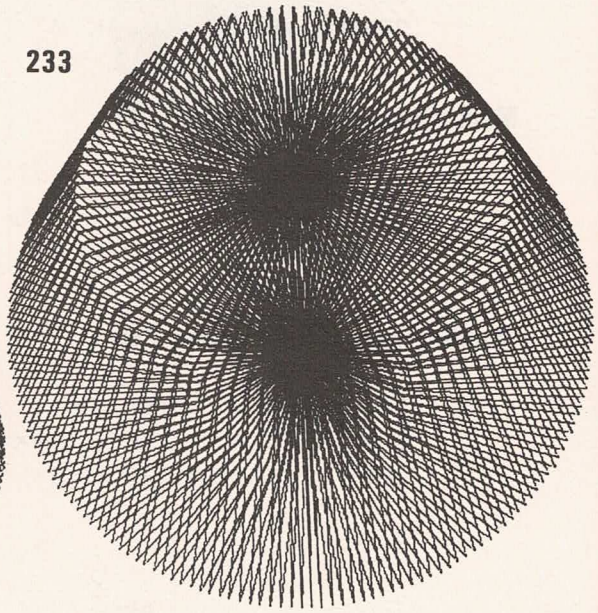


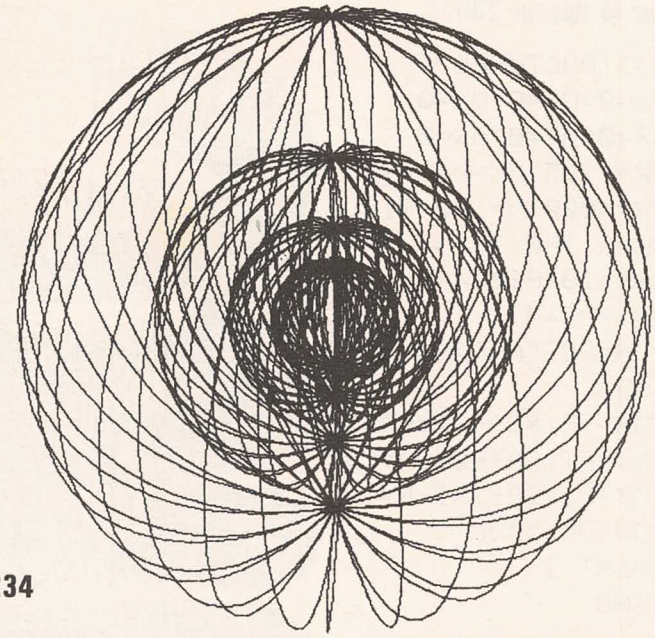
231

232



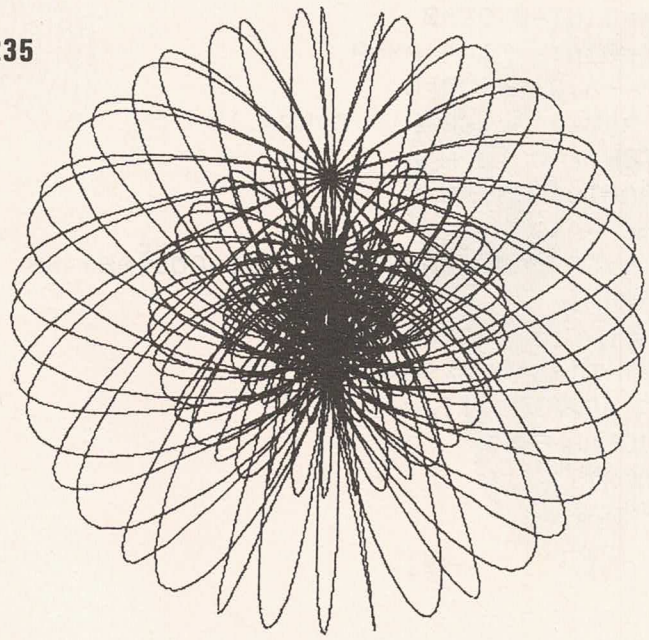
233





234

235

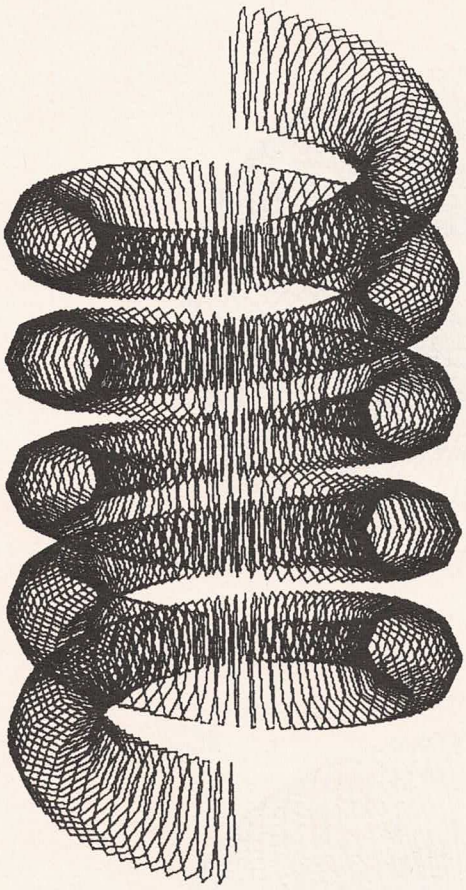


Modifications pour le dessin 236

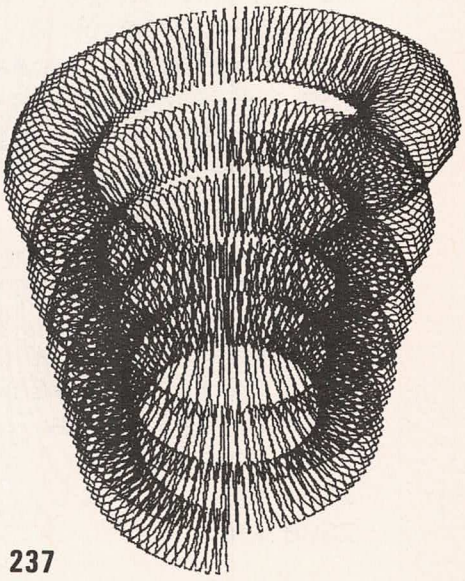
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=0:AX=0
130 QX=-2.5:QY=0:QZ=0
2000 N=4000:M=100:R1=.6:R2=.15
2100 FOR I=0 TO N
2110 AA=10*PI*I/N
2120 AB=AA*M
2130 CA=COS(AA):SA=SIN(AA):CB=COS(AB):SB
=SIN(AB)
2140 MX=-R1*CA-R2*CA*CB
2150 MY=R1*SA+R2*SA*CB
2160 MZ=R2*SB+2*I/N-1
2200 GOSUB 5000
2300 NEXT I
2500 END
```

Modifications pour le dessin 237

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-PI/4:AX=0
130 QX=-3:QY=0:QZ=0
2000 N=4000:M=100:R1=.6:R2=.15
2100 FOR I=0 TO N
2110 AA=10*PI*I/N
2120 AB=AA*M
2130 CA=COS(AA):SA=SIN(AA):CB=COS(AB):SB
=SIN(AB)
2140 MX=-R1*CA-R2*CA*CB
2150 MY=R1*SA+R2*SA*CB
2160 MZ=R2*SB+2*I/N-1
2200 GOSUB 5000
2300 NEXT I
2500 END
```



236



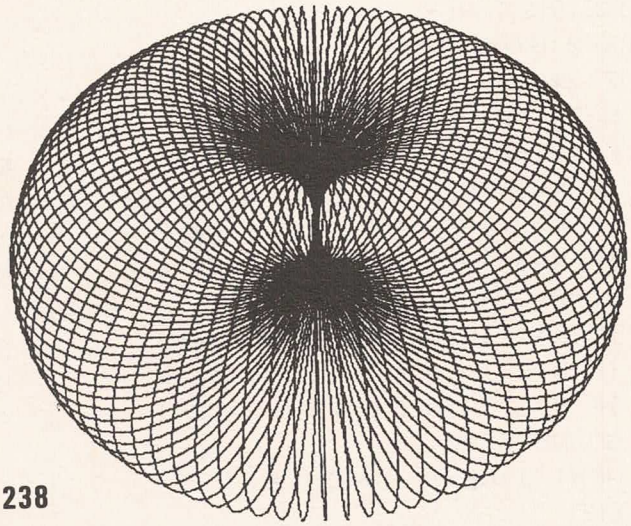
237

Modifications pour le dessin 238

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-PI/4:AX=0
130 QX=-2.5:QY=0:QZ=0
2000 N=4000:M=100:R1=.5:R2=.5
2100 FOR I=0 TO N
2110 AA=2*PI*I/N
2120 AB=AA*M
2130 CA=COS(AA):SA=SIN(AA):CB=COS(AB):SB
=SIN(AB)
2140 MX=-R1*CA-R2*CA*CB
2150 MY=R1*SA+R2*SA*CB
2160 MZ=R2*SB
2200 GOSUB 5000
2300 NEXT I
2500 END
```

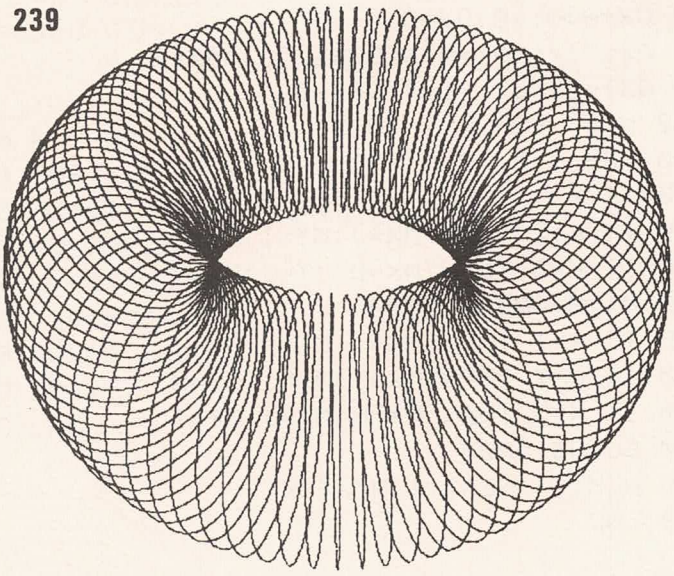
Modifications pour le dessin 239

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-PI/4:AX=0
130 QX=-2:QY=0:QZ=0
2000 N=4000:M=100:R1=.6:R2=.25
2100 FOR I=0 TO N
2110 AA=2*PI*I/N
2120 AB=AA*M
2130 CA=COS(AA):SA=SIN(AA):CB=COS(AB):SB
=SIN(AB)
2140 MX=-R1*CA-R2*CA*CB
2150 MY=R1*SA+R2*SA*CB
2160 MZ=1.5*R2*SB
2200 GOSUB 5000
2300 NEXT I
2500 END
```



238

239



Modifications pour le dessin 240

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=2*PI/5:AY=0:AX=0
2000 N=7: FOR K=1 TO 3
2010 FOR I=0 TO N:FOR J=0 TO N
2100 IF K=1 THEN MX=I:MY=0:MZ=J
2110 IF K=2 THEN MX=J:MY=I:MZ=0
2120 IF K=3 THEN MX=0:MY=J:MZ=I
2150 B1=0:GOSUB 5000
2200 IF K=1 THEN MX=I:MY=N:MZ=J
2210 IF K=2 THEN MX=J:MY=I:MZ=N
2220 IF K=3 THEN MX=N:MY=J:MZ=I
2250 GOSUB 5000
2300 NEXT J:NEXT I:NEXT K
2500 END
```

Modifications pour le dessin 241

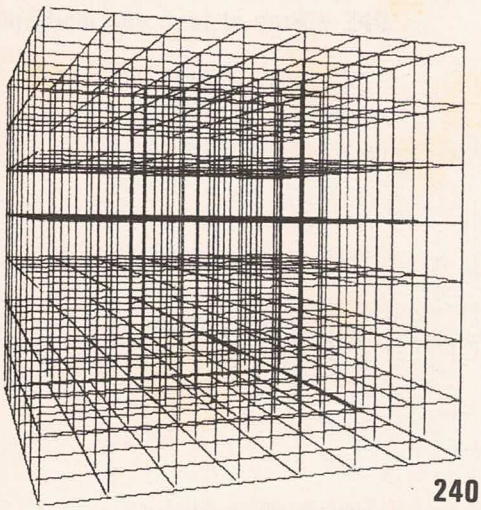
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=PI/4:AY=ATN(SQR(.5)):AX=0
130 QX=-12:QY=-1.4:QZ=.8
2000 N=7: FOR K=1 TO 3
2010 FOR I=0 TO N:FOR J=0 TO N
2100 IF K=1 THEN MX=I:MY=0:MZ=J
2110 IF K=2 THEN MX=J:MY=I:MZ=0
2120 IF K=3 THEN MX=0:MY=J:MZ=I
2150 B1=0:GOSUB 5000
2200 IF K=1 THEN MX=I:MY=N:MZ=J
2210 IF K=2 THEN MX=J:MY=I:MZ=N
2220 IF K=3 THEN MX=N:MY=J:MZ=I
2250 GOSUB 5000
2300 NEXT J:NEXT I:NEXT K
2500 END
```

Modifications pour le dessin 242

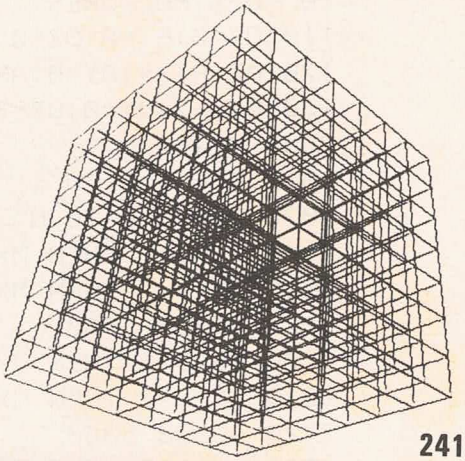
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=PI/4:AY=0:AX=0
130 QX=-2:QY=0:QZ=3
2000 N=100
2010 FOR J=1 TO 2
2020 FOR I=0 TO N :B1=0
2100 IF J=1 THEN MX=I:MY=0:MZ=0
2110 IF J=2 THEN MX=0:MY=I:MZ=0
2120 GOSUB 5000
2200 IF J=1 THEN MX=I:MY=N:MZ=0
2210 IF J=2 THEN MX=N:MY=I:MZ=0
2220 GOSUB 5000
2300 NEXT I:NEXT J
2500 END
```

Modifications pour le dessin 243

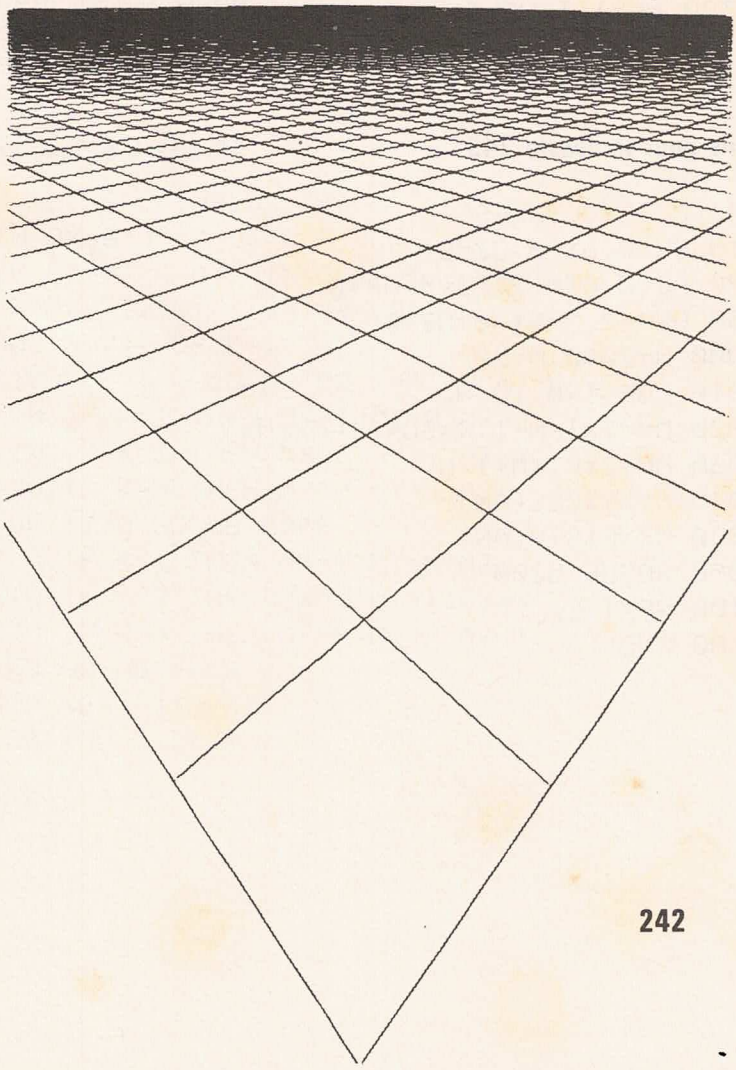
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=2000:M=20
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```



240



241



242

Modifications pour le dessin 244

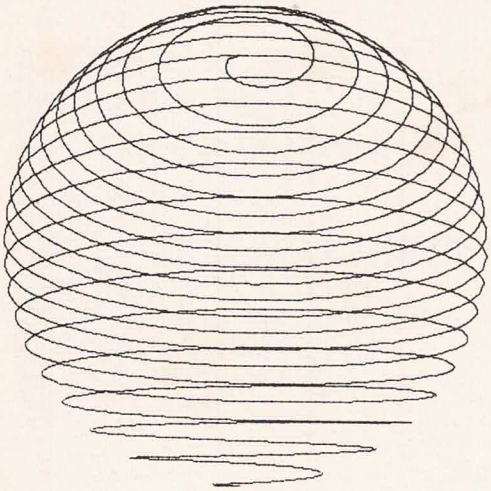
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=2000:M=50
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 245

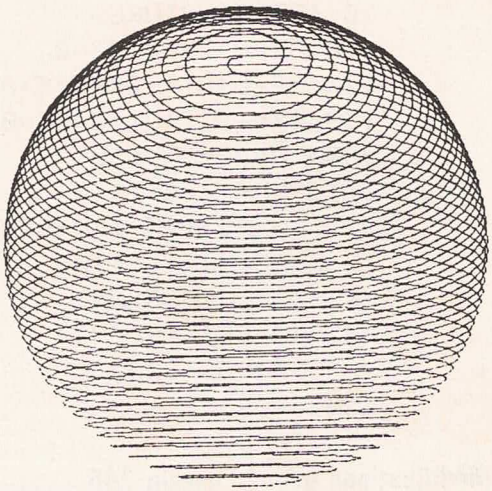
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=2000:M=20
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=1-ABS(MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 246

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=2000:M=50
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=1-ABS(MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

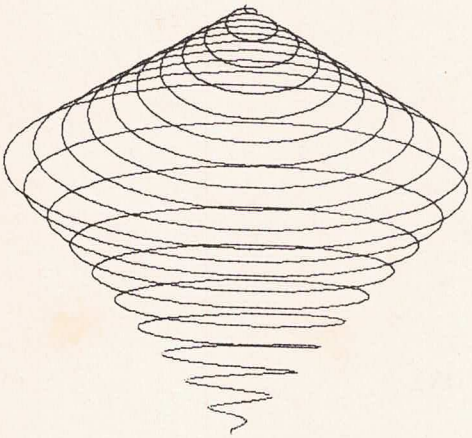


243

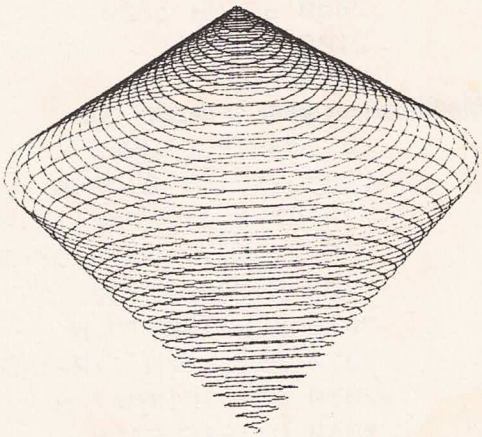


244

245



246



Modifications pour le dessin 247

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=2000:M=200
2010 FOR I=0 TO N/2
2020 MX=2*I/N-1:R=SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 248

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=150:M=50
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 249

```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=-PI/6:AY=0:AX=0
130 QX=-2.7:QY=0:QZ=3
2000 N=200:M=50
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=.5*SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 250

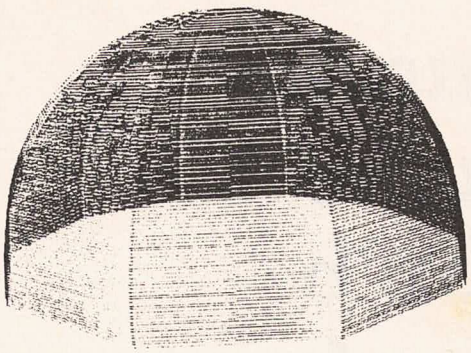
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=-PI/6:AY=-PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=200:M=50
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 251

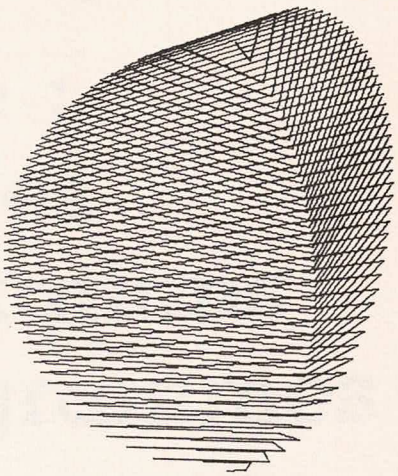
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-2*PI/5:AX=0
130 QX=-2.7:QY=0:QZ=0
2000 N=200:M=50
2010 FOR I=0 TO N
2020 MX=2*I/N-1:R=SQR(1-MX*MX)
2030 AN=2*PI*M*I/N
2040 MY=R*COS(AN)
2050 MZ=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```

Modifications pour le dessin 252

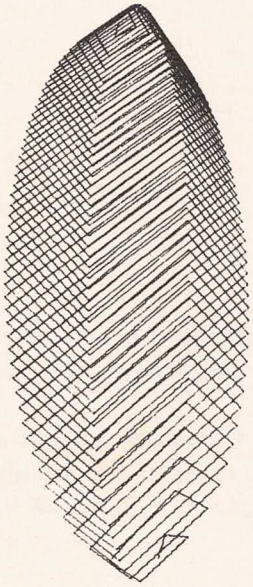
```
10 'D3STRUCTURES
110 OX=0:OY=0:OZ=0
120 AZ=0:AY=-PI/9:AX=0
130 QX=-2.5:QY=0:QZ=0
2000 N=2000:M=100
2010 FOR I=0 TO N
2020 MZ=2*I/N-1:R=.3+.3*SIN(2*PI*MZ+3*PI
/2)
2030 AN=2*PI*M*I/N
2040 MX=R*COS(AN)
2050 MY=R*SIN(AN)
2060 GOSUB 5000
2100 NEXT I
2500 END
```



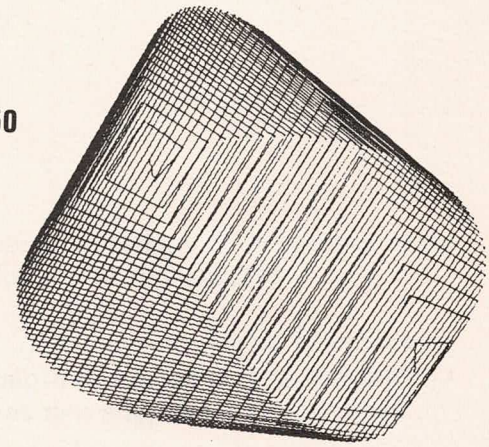
247



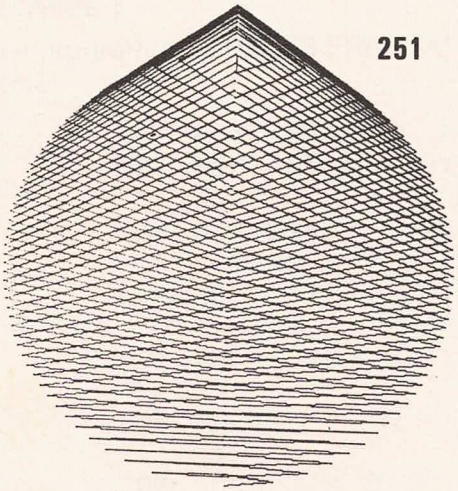
248



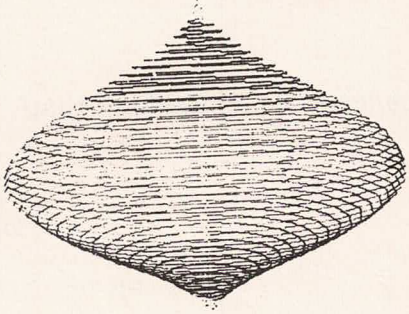
249



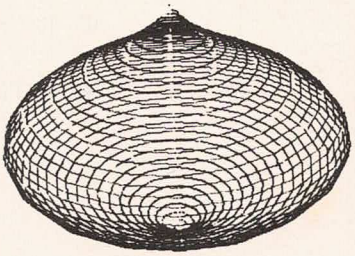
250



251



252



Annexe : exemples d'adaptations de programmes

A titre d'exemples nous donnons ici quelques adaptations de programmes du livre. On constatera que peu de modifications sont nécessaires.

Je tiens à remercier Bernard Doucy (Becy Informatique, 78 bd Montebello, Lille), Alain Juhel et Jean Kiszel qui ont accepté de me prêter leur matériel ou qui ont réalisé et testé eux-mêmes certaines des adaptations proposées.

Exemple 1

Le programme ÉTOILES FRACTALES pour l'Apple II, dessin sur le moniteur-télé.

```
10  REM :ETOILES FRACTALES
50  HGR :NP = 159:PI =  ATN (1) *
     4
100 N = 5:K = 5:RA = .35:LL = NP *
    .8:AA = 4 * PI / 5
110 X0 = (NP - LL) / 2:Y0 = NP *
    .4:A0 =  - AA
120 XW =  INT (X0):YW =  INT (Y0)

190 NN = N * (N - 1) * (K - 1) -
     1
200  FOR I = 0 TO NN
```

```
210 I1 = I:H = 0
300  IF (I1 -  INT (I1 / (N - 1))
     * (N - 1)) = 0 AND H ( K -
     1 THEN I1 = I1 / (N - 1):H =
     H + 1: GOTO 300
310 L0 = LL * RA ↑ (K - 1 - H)
320 A0 = A0 + AA
330 X0 = X0 + L0 *  COS (A0)
340 Y0 = Y0 + L0 *  SIN (A0)
350  HPLOT XW,YW TO  INT (X0), INT
     (Y0):XW =  INT (X0):YW =  INT
     (Y0)
400  NEXT I
1000  END
```

Exemple 2

Un programme ÉTOILES FRACTALES, pour l'Apple II, dessin sur un traceur « MCP-40 4 colours Printer/Plotter ».

```
10  REM :ETOILES FRACTALES
50  PR# 1: PRINT  CHR$ (18): PRINT
     "M0,-500": PRINT "I":NP = 48
     0:PI =  ATN (1) * 4
100 N = 5:K = 5:RA = .35:LL = NP:
     AA = 4 * PI / 5
110 X0 = (NP - LL) / 2:Y0 = NP /
     4:A0 =  - AA
120  PRINT "M"; INT (X0);","; INT
     (Y0)
190 NN = N * (N - 1) ↑ (K - 1) -
     1
200  FOR I = 0 TO NN
210 I1 = I:H = 0
300  IF (I1 -  INT (I1 / (N - 1))
     * (N - 1)) = 0 AND H ( K -
     1 THEN I1 = I1 / (N - 1):H =
     H + 1: GOTO 300
310 L0 = LL * RA ↑ (K - 1 - H)
320 A0 = A0 + AA
330 X0 = X0 + L0 *  COS (A0)
340 Y0 = Y0 + L0 *  SIN (A0)
350  PRINT "D"; INT (X0);","; INT
     (Y0)
400  NEXT I
1000  END
```

Exemple 3

Le programme DRAGONS pour l'Apple II, dessin sur tracteur « MCP-40 4 colours Printer/Plotter ».

```
10  REM :DRAGONS
50  PR# 1: PRINT  CHR$ (18): PRINT
    "M0,-800": PRINT "I":NP = 48
    0:PI =  ATN (1) * 4
100 N = 10
110  DIM A(N)
120  FOR I = 1 TO N:A(I) = 0: NEXT
    I
130 X0 = NP / 3:Y0 = NP / 2:A0 =
     - PI / 4 * (N - 2):L0 = NP /
     SQR (2) ^ N
140 X1 = X0:Y1 = Y0:X2 = X0:Y2 =
    Y0
150  PRINT "M"; INT (X0);","; INT
    (Y0)
190 NN = 2 ^ N - 1
200  FOR I = 0 TO NN
210  IF I = 0 THEN 300
215 II = I:J = 0
220  IF II -  INT (II / 2) * 2 =
    0 THEN II = II / 2:J = J + 1
    : GOTO 220
230 AA = (A(N - J) * 2 - 1) * (((
    II - 1) / 2 -  INT ((II - 1)
     / 2 / 2) * 2) * 2 - 1) * PI
     / 2
240 A0 = A0 + AA
300 X0 = X1:Y0 = Y1:X1 = X2:Y1 =
    Y2:X2 = X2 + L0 *  COS (A0):
    Y2 = Y2 + L0 *  SIN (A0)
310  PRINT "D"; INT ((X0 + 3 * X1
    ) / 4);","; INT ((Y0 + 3 * Y
    1) / 4)
315  PRINT "D"; INT ((X2 + 3 * X1
    ) / 4);","; INT ((Y2 + 3 * Y
    1) / 4)
320 X0 = X1:Y0 = Y1
330  NEXT I
1000  END
```

Exemple 4

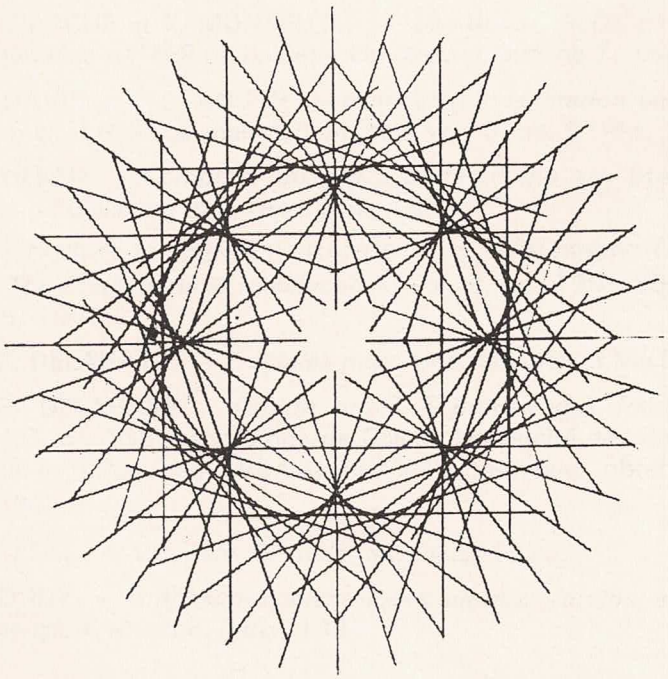
Le programme LINÉAIRES BÂTONS pour l'ORIC ou l'ATMOS, dessin sur le moniteur-télé.

```
10 REM LINEAIRES BATONS
50 HIRES: NP=180
100 N=100: M=1: K=5
200 FOR I=1 TO M
210 R1=NP/4: R2=NP*5/24
300 FOR J=0 TO N-1
310 AN=2*PI*J/N
320 XD=NP/2+R1*COS( AN) +R2*COS( K*AN)
330 YD=NP/2+R1*SIN( AN) +R2*SIN( K*AN)
340 XA=NP/2+R1*COS( AN) +R2*COS( K*AN+PI)
350 YA=NP/2+R1*SIN( AN) +R2*SIN( K*AN+PI)
400 CURSET XD, YD, 1
410DRAW XA-XD, YA-YD, 1
500 NEXT J: NEXT I
1000 END
```

Exemple 5

Le programme LINÉAIRES BÂTONS pour le Sharp MZ 730, dessin sur le traceur.

```
10 REM LINEAIRES BATONS
50 MODE GR: MOVE 0,-500:HSET:NP=480:PI=A
TN(1)*4
100 N=300:M=4:K=3
200 FOR I=1 TO M
205 PCOLORI-1
210 R1=NP/3*.6↑(I-1):R2=NP/8*.6↑(I-1)
300 FOR J=0 TO N-1
310 AN=2*J*PI/N
320 XD=NP/2+R1*COS(AN)+R2*COS(K*AN)
330 YD=NP/2+R1*SIN(AN)+R2*SIN(K*AN)
340 XA=NP/2+R1*COS(AN)+R2*COS(K*AN+PI)
350 YA=NP/2+R1*SIN(AN)+R2*SIN(K*AN+PI)
400 MOVE XD,YD
410 LINE XA,YA
500 NEXT J:NEXT I
1000 END
```



Exemple 6

Le programme LINÉAIRES BÂTONS pour le SIL'Z Leanord, dessin sur le moniteur-télé.

```
10 'LINEAIRES BATONS
20 X$=FNSCNGRA$
30 X$=FNCLRSCN$(1)
50 PI=ATN(1)*4:NP=256:C=1.7
100 N=100:M=1:K=5
200 FOR I=1 TO M
210 R1=NP/4:R2=NP*5/24
300 FOR J=0 TO N-1
310 AN=2*J*PI/N
320 XD=C*(NP/2+R1*COS(AN)+R2*COS(K*AN))+50
330 YD=NP/2+R1*SIN(AN)+R2*SIN(K*AN)
340 XA=C*(NP/2+R1*COS(AN)+R2*COS(K*AN+PI))+50
350 YA=NP/2+R1*SIN(AN)+R2*SIN(K*AN+PI)
400 X$=FNTRCSEG$(INT(XD),INT(YD),INT(XA),INT(YA),0,0)
500 NEXT J:NEXT I
1000 END
```

Références bibliographiques

[1] • J. BRETTE. — *Courbes Mathématiques.* Numéro spécial de la Revue du Palais de la Découverte, Paris, 1976.

[2] • M. DARCHE et P. MONSELLIER. — *Les gones.* PLOT n° 10 (Bulletin des Régionales APMEP de Poitiers, Limoges et Orléans-Tours), 1980.

[3] • C. DAVIS et D.E. KNUTH. — *Number representation and dragon curves.* Journal of Recreational Mathematics Vol. 3, pp. 61-81, 133-149, 1970.

[4] • J. DEKEN. — *Computer Images : State of the art.* Steward Tabori and Chang Publishing Inc., New York, 1983.

[5] • M. DEKKING, M. MENDES-FRANCE and A. VAN DER PORTEN. — *Folds I, II, III. The Mathematical Intelligences,* vol. 4, pp. 130-139, 173-181, 190-195, 1983.

[6] • J.-P. DELAHAYE. — *Fractales pour tous.* Science et Vie Micro. Mai 1984.

[7] • J.-P. DELAHAYE. — *Quatre notes à propos des fractales.* Publication A.N.O. n° 138. Laboratoire de Calcul, Université des Sciences et Techniques de Lille. Juin 1984 (écrire à l'auteur pour obtenir un exemplaire gratuit).

[8] • G. DIEHL. — *Vasarely.* Flammarion, Paris, 1979.

[9] • R. DONY. — *Graphismes scientifiques sur micro-ordinateur : de la 2e à la 3e dimension.* Masson, Paris, 1984.

[10] • D. DUBANT. — *Arts ou Mathématiques : la troisième dimension sur P.C. 1500.* L'Ordinateur de Poche n° 55, 1983.

[11] • B. ERNST. — *The magic mirror of M.C. Escher.* Ballantine Books, New York, 1976.

[12] • W. FARNWORTH. — *Fils et formes.* Dessain et Tolra, Paris, 1979.

[13] • H.V. FRANKE. — *Computer Grafik, Galerie.* Du Mont Buchverlag, Köln, 1984.

[14] • Y. GARDAN et M. LUCAS. — *Techniques graphiques interactives et C.A.O.* Hermes Publishing (France), Paris, 1983.

[15] • M. GARDNER. *La courbe du dragon,* Math. Festival, Bibliothèque pour la Science, Belin, Paris, 1981.

[16] • D. GREENBERG, A. MARCUS, A. H. SCHMIDT, V. GORTER, *The computer image : Application of computer graphics.* Addison-Wesley Publishing Company. Reading, 1982.

[17] • M. HALLAIN. *Vasarely.* Fernand Hazan Editeur, Paris, 1973.

[18] • J.-D. LAWRENCE. *A catalogue of special plane curves.* Dover Publications Inc., New York, 1972.

[19] • R. LEAVIT. *Artist and computer.* Harmony Press, New York, 1976.

[20] • H. LEHNING et D. JAKUBOWICZ. *Mathématiques par l'informatique individuelle.* Tome 4 : graphisme. Masson, Paris, 1984.

[21] • J.-L. LOCHER. *Le monde de M.-C. Escher.* Editions du Chêne, Paris, 1972.

[22] • J.-L. LOCHER. — *La vie et l'œuvre de M.-C. Escher.* Chêne/Hachette, Paris, 1981.

[23] • E.-H. LOCKWOOD. — *A book of curves.* Cambridge at the University Press, Cambridge, 1961.

[24] • B. MANDELBROT. — *Les objets fractals.* La Recherche n° 85, 1978.

[25] • B. MANDELBROT. — *The fractal Geometry of nature.* W.A. Freeman and Company, San Francisco, 1983.

[26] • G. MARSHALL. — *Programmer vos graphismes sur micro-ordinateur.* Les Editions d'Organisation, Paris, 1984.

[27] • W.-M. NEWMAN and R.-F. SPROULL. — *Principles of interactive Computer Graphics,* 2^{d} Edition. Mac Graw-Hill, New York, 1979.

[28] • T.-J. O'MALLEY. — *25 programmes graphiques en Basic microsoft.* Tab. Books Inc., Philadelphia, 1983, et Eyrolles, Paris, 1985.

[29] • M.-L. PRUEITT. — *Computer Graphics : 118 computer - generated Designs.* Dover Publications Inc., New York, 1975.

[30] • G. PEANO. *Sur une courbe plane qui remplit toute une aire plane.* Mathematische Annalen, n° 36, 1890.

[31] • P.-S. STEVENS. — *Handbook of Regular Patterns.* The MIT Press, Cambridge, 1981.

[32] • P. TOUGNÉ. — *Pliages de papier, courbes de Dragon, dimension et entropie.* Pour la Science, Février 1984.

[33] • VASARELY. — *Gea.* Editions Hervas, Paris, 1982.

[34] • VON KOCH. — *Sur une courbe continue sans tangente obtenue par une construction géométrique élémentaire.* Arkiv for Matematik Astronomi och Fysik, p. 681-702, 1904.

[35] • J.-L. VULDY. — *Graphisme 3D sur votre micro-ordinateur.* Editions Eyrolles. Paris, 1983.

[36] • Z. — *Les bolygones.* Le petit Archimède n° 27-28, 1976.

]37] • Z. — *Les jolygones.* Le petit Archimède n° 14, 1974.

[38] • Z. — *Dossier Arts et Informatique.* Education et Informatique n° 9, 1982.

[39] • Z. — *Malmö Konsthall Katalog nr 63 : Vasarely.* Repo Helsingborg, 1980.

[40] • Z. — *Générations Infographiques : images du futur.* Catalogue de l'exposition réalisée du 7 juillet au 6 septembre 1982 par l'Atelier de Recherches Techniques Avancées (A.R.T.A.) du Centre Georges Pompidou, Paris, 1982.

Références à consulter selon les chapitres et selon les sujets

Chapitre 1 : [2] [9] [20] [36] [37]
Chapitre 2 : [11] [21] [22] [31]
Chapitre 3 : [3] [5] [15] [25] [32]
Chapitre 4 : [6] [7]
Chapitre 5 : [1] [18] [23]
Chapitre 6 : [12]
Chapitre 7 : [6] [7] [24] [25] [30] [34]
Chapitre 8 : [8] [17] [33] [39]
Chapitre 9 : [9] [10] [20] [29]
Chapitre 10 : [9] [20] [28] [35]
Art et Ordinateurs : [4] [13] [16] [19] [29] [38] [40]
Graphismes et Ordinateurs : [9] [14] [16] [20] [26] [27] [28]).

Imprimé en France. — JOUVE, 18, rue Saint-Denis, 75001 PARIS
N° 13851. Dépôt légal : Janvier 1985
N° d'Editeur : 4202

Presque tous les ordinateurs individuels disposent aujourd'hui de possibilités graphiques qui permettent de réaliser d'extraordinaires dessins géométriques et artistiques.

Le but de ce livre est de vous apprendre à obtenir de tels dessins. Il comporte de nombreux exemples (plus de 250) avec, pour chacun d'eux, le programme correspondant. Les programmes sont rédigés en BASIC et de manière à pouvoir être utilisés sur *tout micro-ordinateur* (moyennant parfois des petites adaptations expliquées dans le texte).

L'utilisation de ce livre ne nécessite pas de connaissances en mathématiques approfondies, ni de connaissances informatiques spécialisées. Ce livre peut même servir de premier livre d'informatique, juste après une initiation au BASIC.

Au cours des chapitres, vous rencontrerez des polygones et des étoiles ; des motifs figuratifs d'Escher qui s'emboîtent miraculeusement ; des fractales (objets infiniment découpés et ramifiés) ; des courbes extraordinaires ; des quadrillages déformés « à la Vasarely » ; de merveilleuses surfaces mathématiques ; et enfin des objets en trois dimensions autour desquels vous pourrez vous déplacer.

Mais le plus grand intérêt de ce livre réside peut-être dans les possibilités créatrices qu'il ouvre. En effet, les programmes sont présentés de façon à pouvoir engendrer de nombreux dessins différents de ceux donnés en exemples. Avec un peu de patience et d'expérience, on sait utiliser ces programmes (et en faire d'autres), ce qui permet de réaliser ses propres créations géométriques et artistiques.

EYROLLES

8224